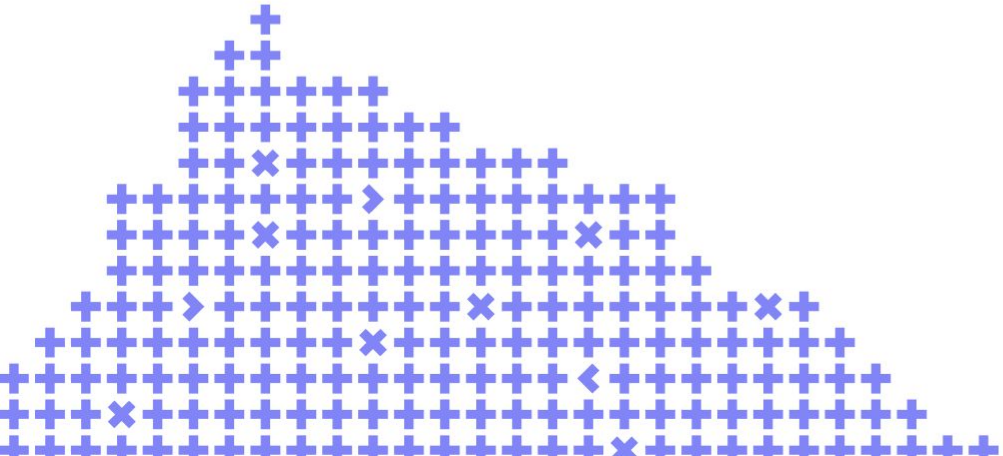
KubeVirt, its networking, and how we brought it to the next level

Andrei Kvapil PALARK



High Load Armenia

Co-organizer

Yandex

About me

Andrei Kvapil
Solutions Architect

andrei.kvapil@palark.com

PALARK

What I do

Architectural solutions based on Kubernetes
Design and developing cloud platforms
Software-defined storage

My experience

2017—2021 — Cloud Architect, WEDOS
since 2021 — Solutions Architect, Palark

The tools I use the most

kubernetes KubeVirt LINSTOR

PALARK

Holistic DevOps & SRE experts

We are

- Germany based company
- Certified Kubernetes Administrators
- Kubernetes organisation members
- Open Source projects creators
- CNCF projects contributors

Our team is

- Building and supporting Linux-based infrastructure since 2008
- Consulting on your app's design to run them in K8s
- Using Kubernetes in production since 2017
- Bringing DevOps & SRE practices with 24/7 support & SLA

We have customers from

- Online retail
- Fintech
- Blockchain
- Online media
- Edtech
- Advertising

Read our tech blog!

Best practices for deploying highly available app in Kubernetes. Part 1

Making the most out of Helm templates

Just-for-fun experiment: Deploying Kubernetes on two old laptops with Gentoo Linux

blog.palark.com

++ High Load Armenia

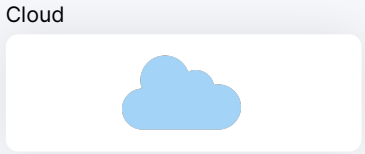
We need a cloud

We need a cloud

Cloud

We need a cloud

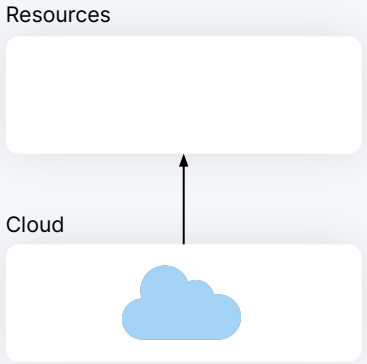
How it should work:



We need a cloud

How it should work:

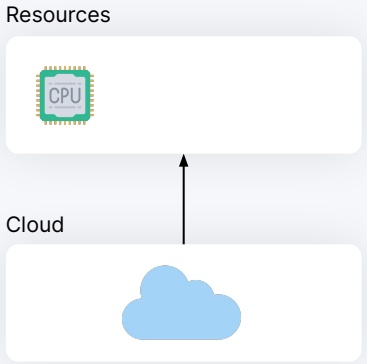
Take physical resources of Nodes



We need a cloud

How it should work:

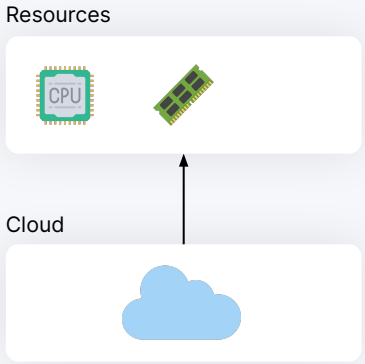
Take physical resources of Nodes



We need a cloud

How it should work:

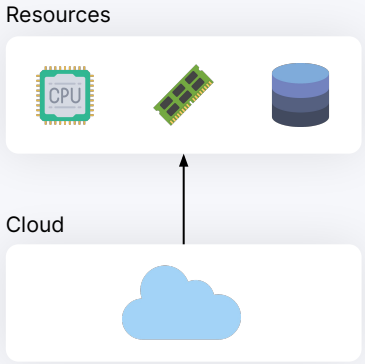
Take physical resources of Nodes



We need a cloud

How it should work:

Take physical resources of Nodes

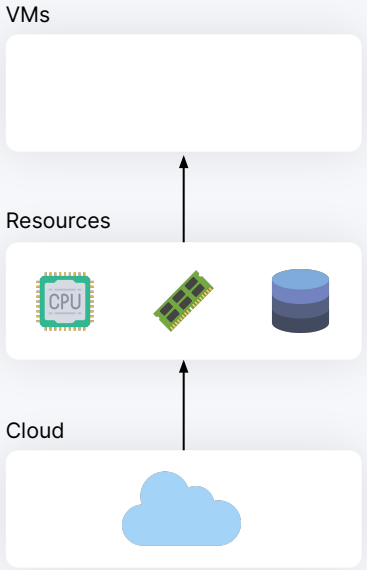


We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

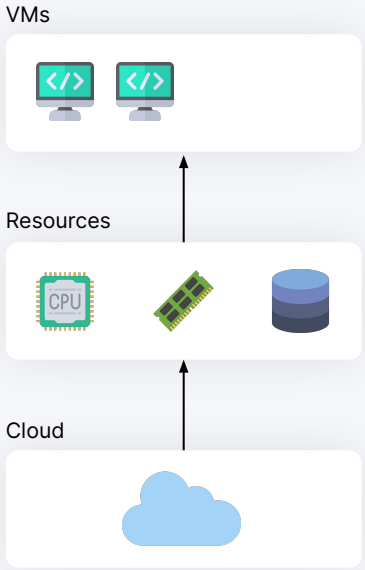


We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

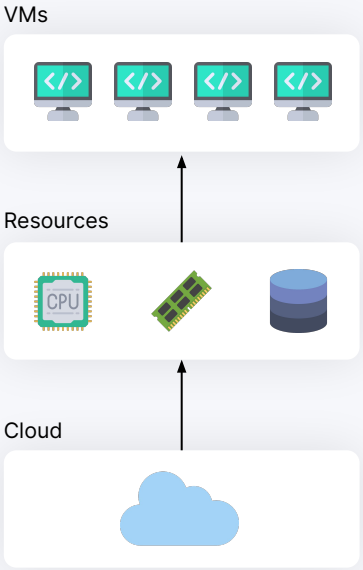


We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them



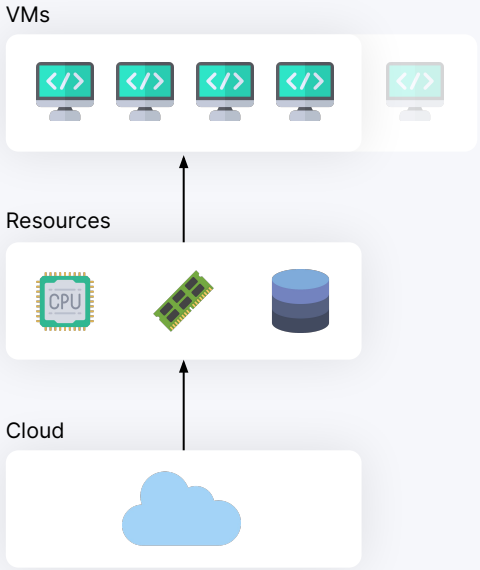
We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:



We need a cloud

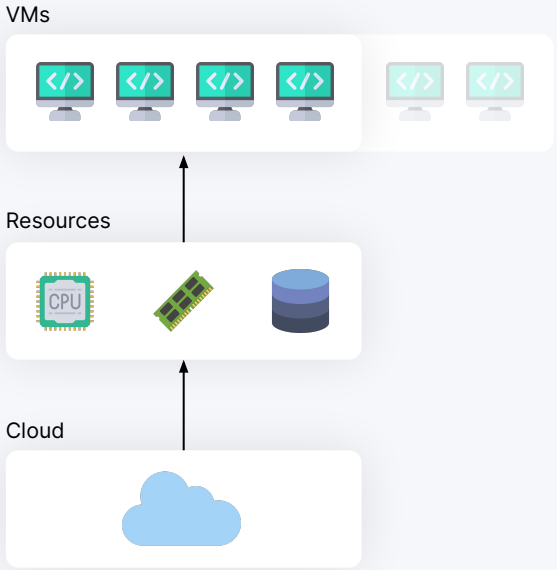
How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs



We need a cloud

How it should work:

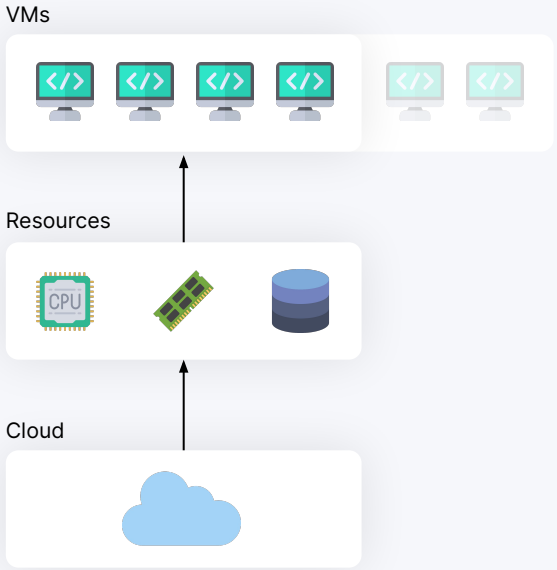
Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:



We need a cloud

How it should work:

Take physical resources of Nodes

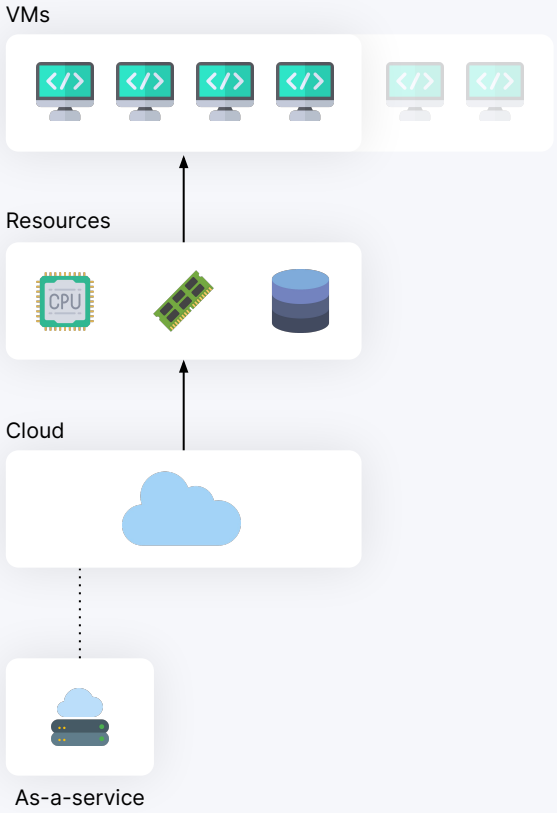
Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

As-a-service



We need a cloud

How it should work:

Take physical resources of Nodes

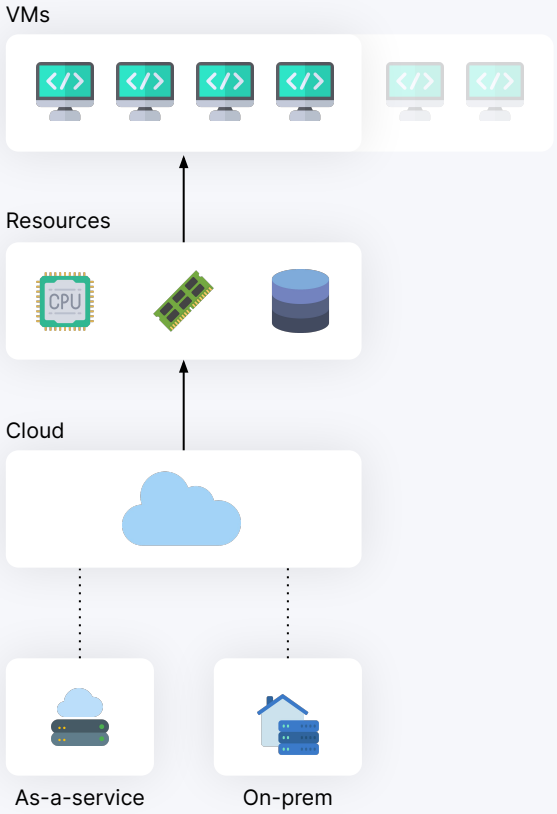
Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

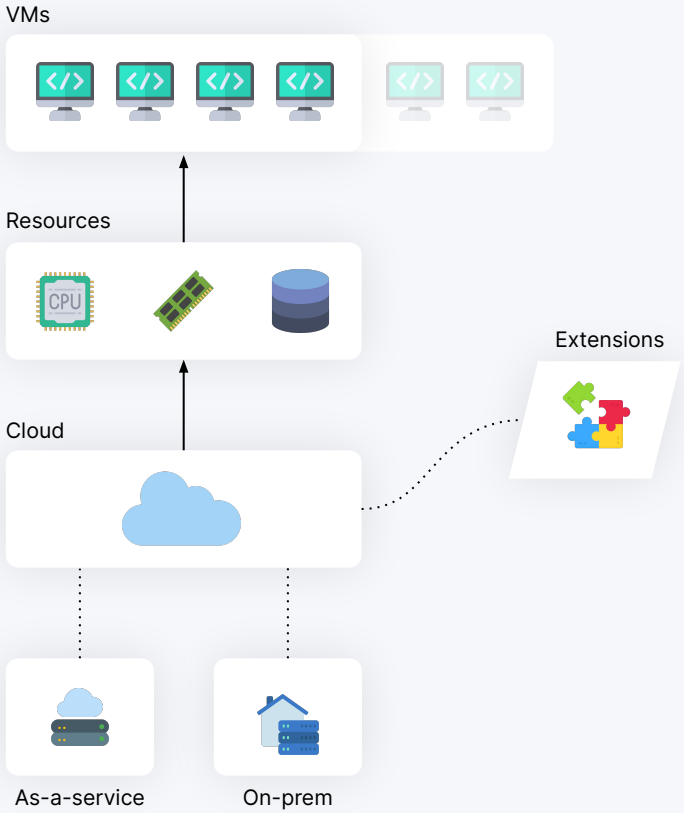
Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem

Has pluggable interface



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

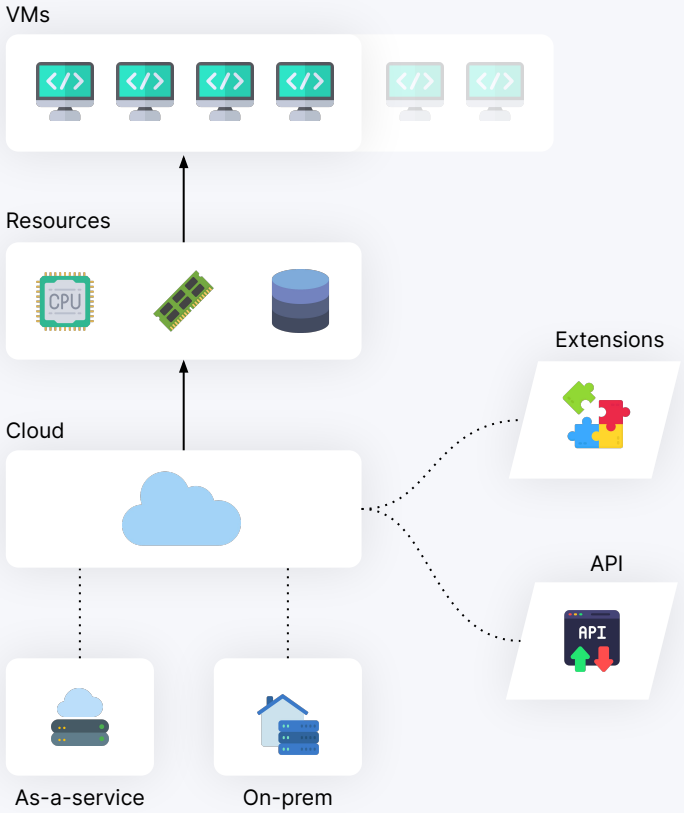
Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem

Has pluggable interface

Provides well-known API



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

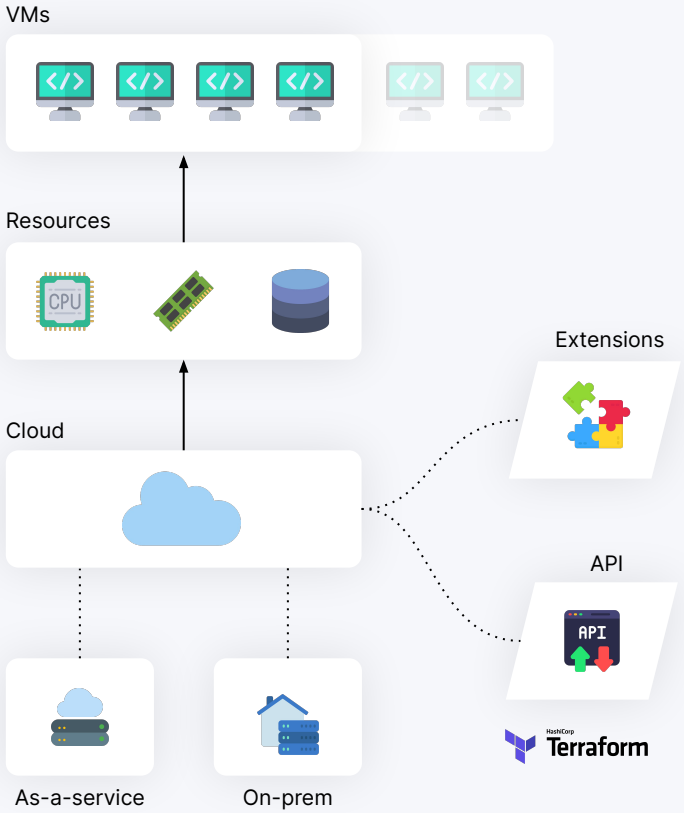
Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem

Has pluggable interface

Provides well-known API



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

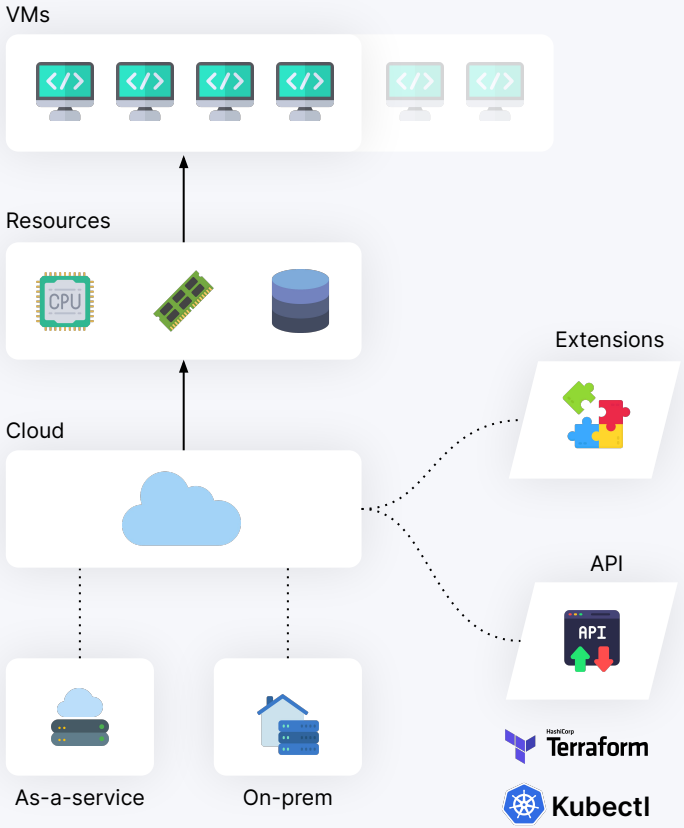
Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem

Has pluggable interface

Provides well-known API



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

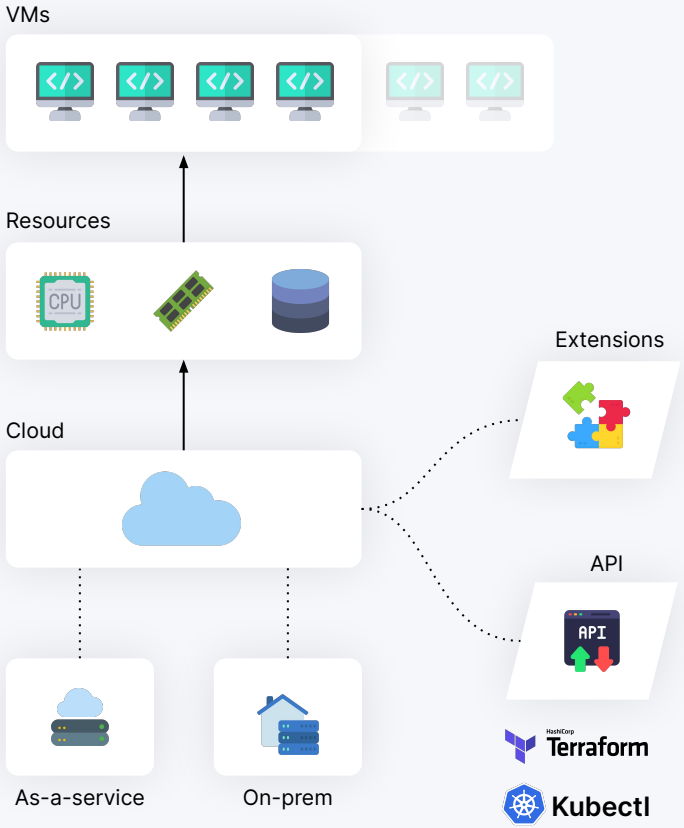
Works:

As-a-service On-prem

Has pluggable interface

Provides well-known API

Consumers:



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

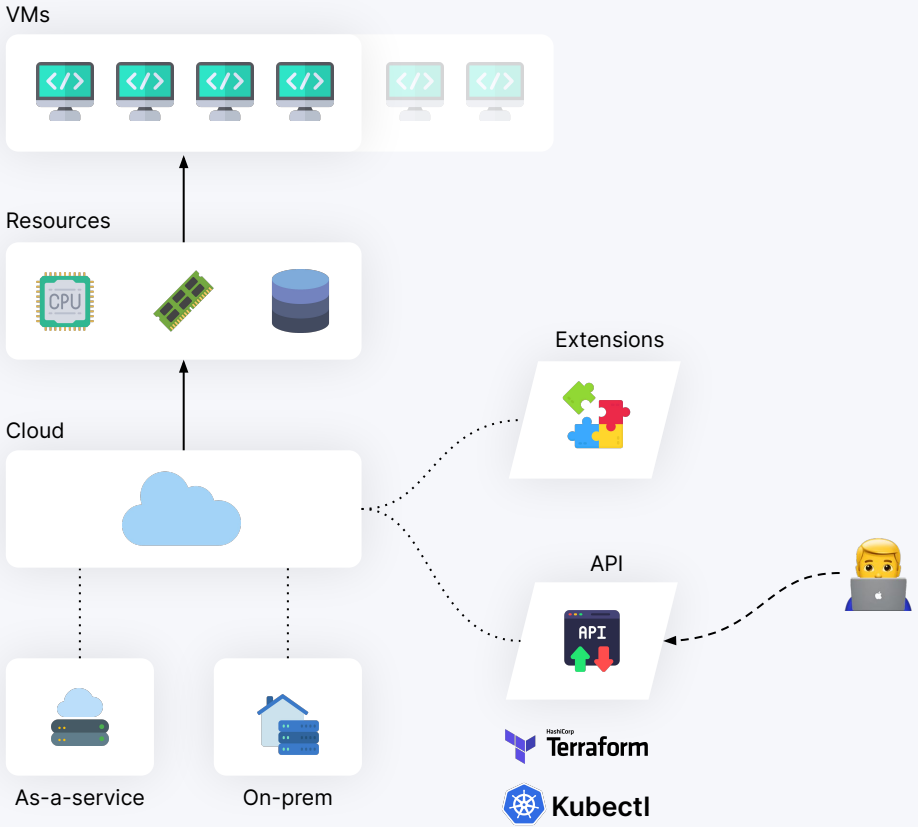
As-a-service On-prem

Has pluggable interface

Provides well-known API

Consumers:

Users



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

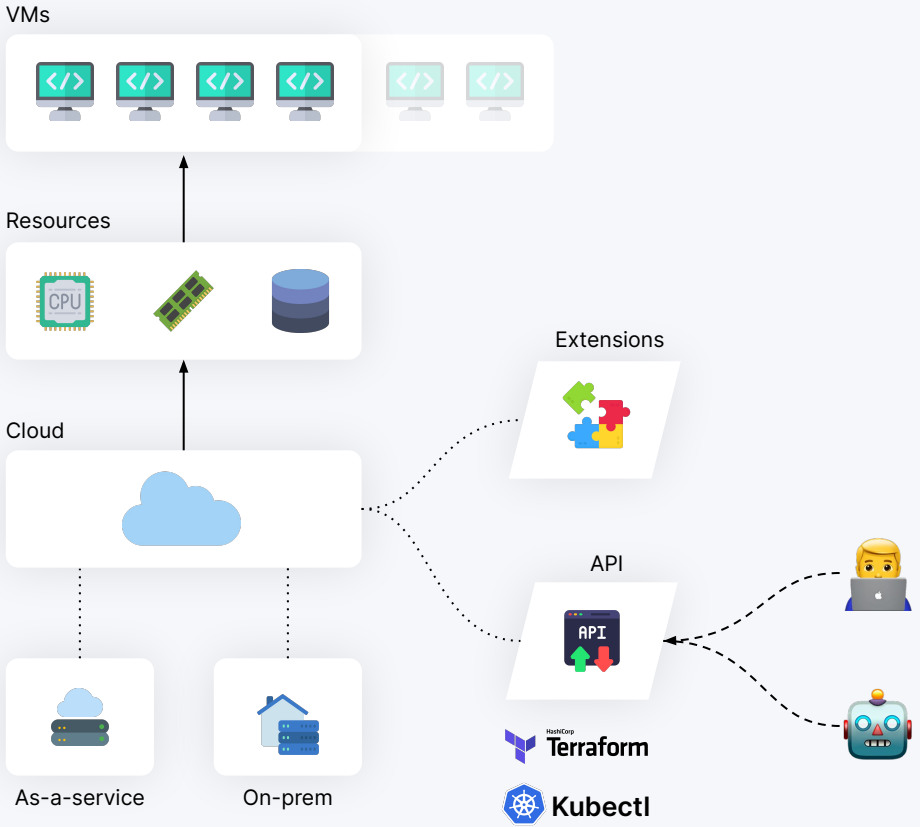
As-a-service On-prem

Has pluggable interface

Provides well-known API

Consumers:

Users Automations



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem

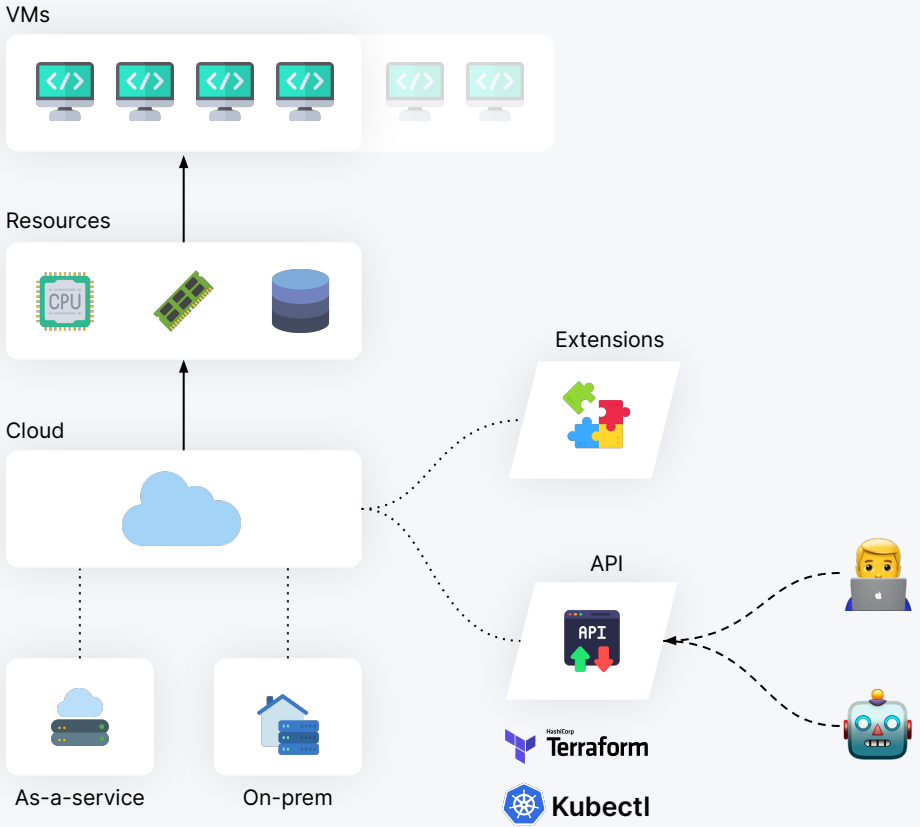
Has pluggable interface

Provides well-known API

Consumers:

Users Automations

Must be Open Source!



We need a cloud

How it should work:

Take physical resources of Nodes

Run VMs on top of them

Be scalable to run:

Legacy software Windows Kubernetes VMs

Works:

As-a-service On-prem

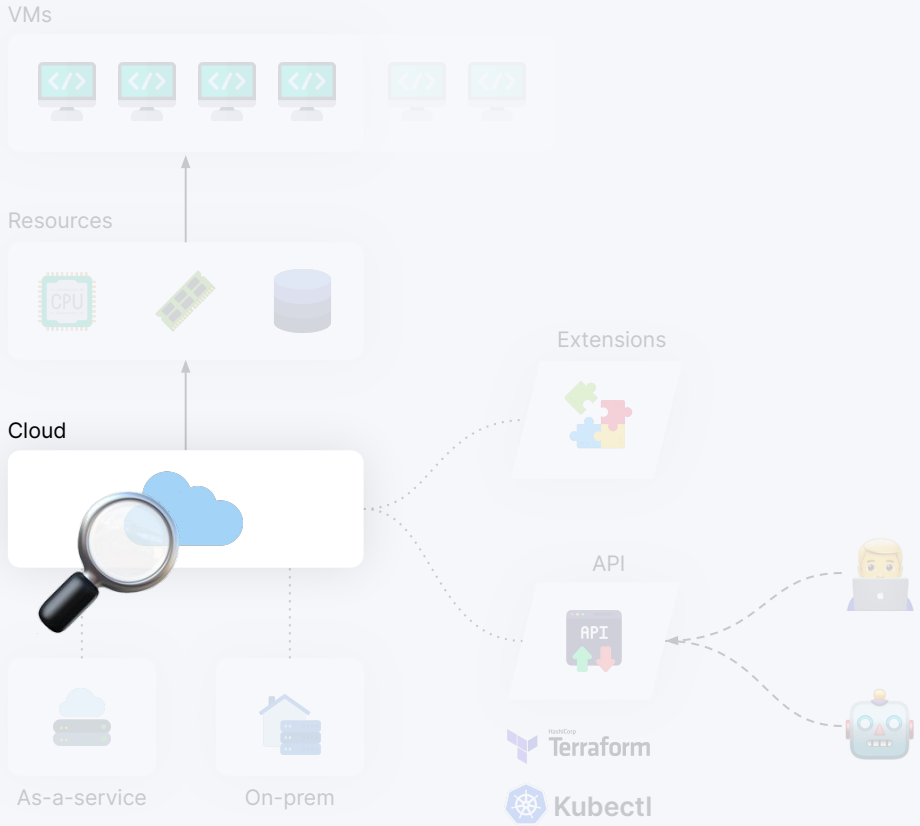
Has pluggable interface

Provides well-known API

Consumers:

Users Automations

Must be Open Source!



Time to choose

Time to choose

Time to choose

Time to choose

Time to choose

Time to choose

Time to choose

Time to choose

👎 Small community or controlled by single company

Time to choose

👎 Small community or controlled by single company

👎 Poor Kubernetes integration

Time to choose

Time to choose

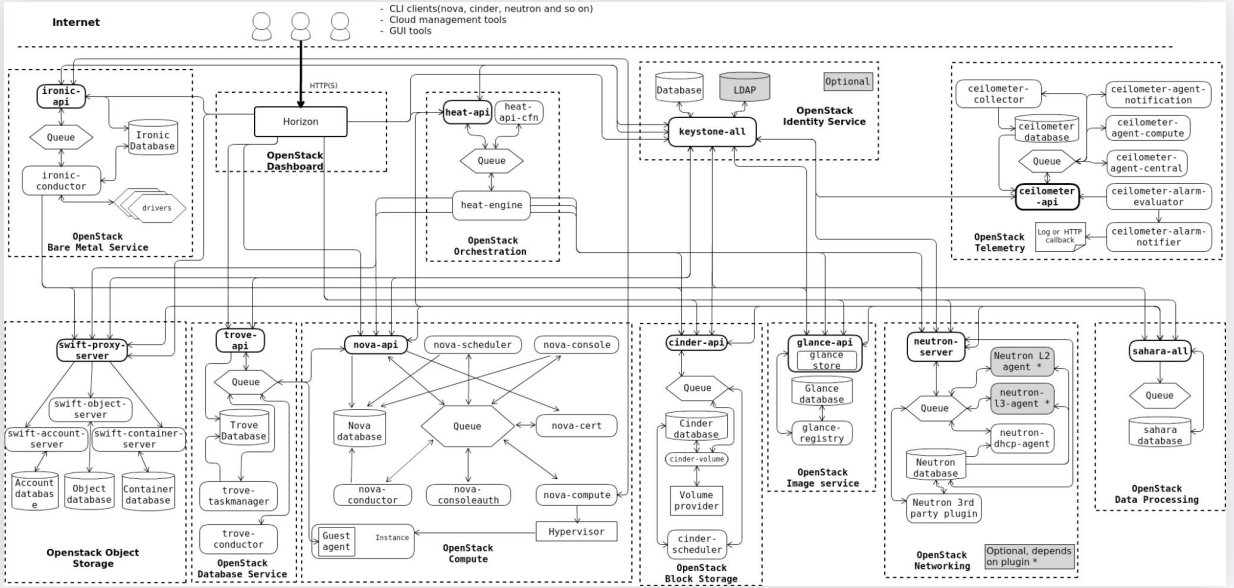
Time to choose

👎 Overcomplicated achitecture

Time to choose

KubeVirt

openstack.



Time to choose

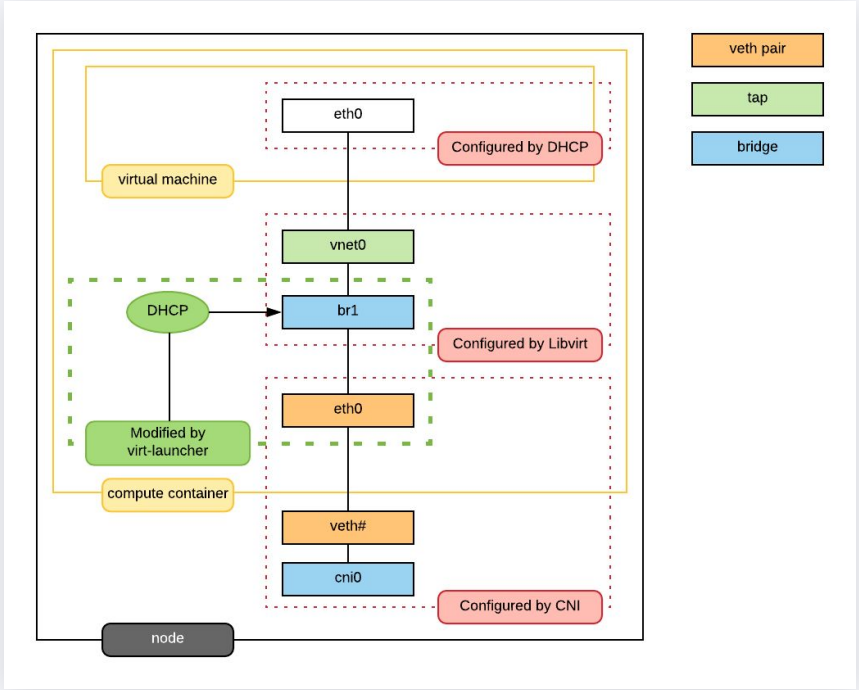
Time to choose

KubeVirt

👎 Overcomplicated networking

Time to choose

KubeVirt



Time to choose

Time to choose

Time to choose

Own solution?

Time to choose

Own solution?

Kubernetes

Time to choose

Own solution?

libvirt

Kubernetes

Time to choose

Own solution?

Container

libvirt

Kubernetes

Time to choose

Own solution?

Container

libvirt

Kubernetes

Time to choose

KubeVirt 👀

Own solution?

Container

libvirt

Kubernetes

Time to choose

Time to choose

KubeVirt

👍 Kubernetes based

Time to choose

👍 Kubernetes based

👍 Growing community

Time to choose

KubeVirt

👍 Kubernetes based

👍 Growing community





Guthub Stars/Forks	3200/707	2800/2500	1200/909	870/402	362/95
GitHub created at	2016.12.16	2010.07.22	2013.04.29	2011.03.10	2014.10.13
Contributors	220	1097	354	128	77

Time to choose

👍 Kubernetes based

👍 Growing community

Time to choose

KubeVirt

- Kubernetes based
- Growing community
- Big players

Red Hat
NVIDIA
arm
SUSE
PLATFORM9

Red Hat
IBM
MIRANTIS
Hewlett Packard Enterprise
intel
HUAWEI
rackspace technology
hp
vmware

HUAWEI
DELL
CITRIX
NOKIA
verizon

Open Nebula
StorPool DISTRIBUTED STORAGE
And some European hostings

Google
Will be handed over to community

Time to choose

- 👍 Kubernetes based
- 👍 Growing community
- 👍 Big players

Time to choose

- 👍 Kubernetes based
- 👍 Growing community
- 👍 Big players
- 👍 Community driven development (no vendor lock-in)

Time to choose

- 👍 Kubernetes based
- 👍 Growing community
- 👍 Big players
- 👍 Community driven development (no vendor lock-in)
- 👍 Simple architecture

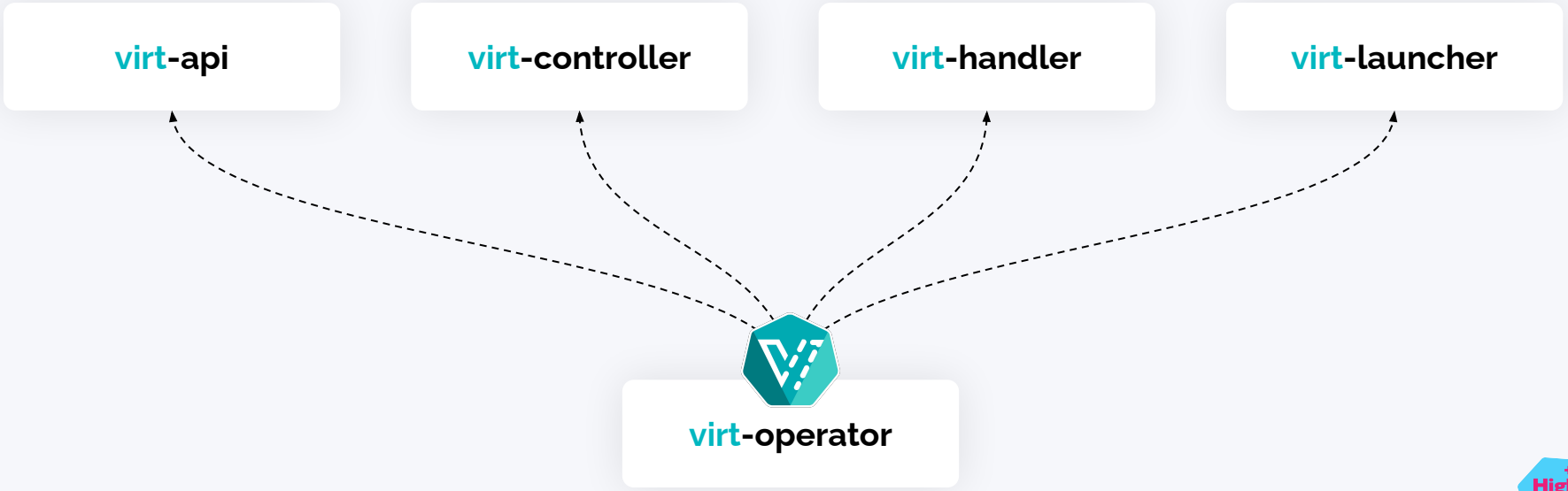
What is KubeVirt?

What is KubeVirt?

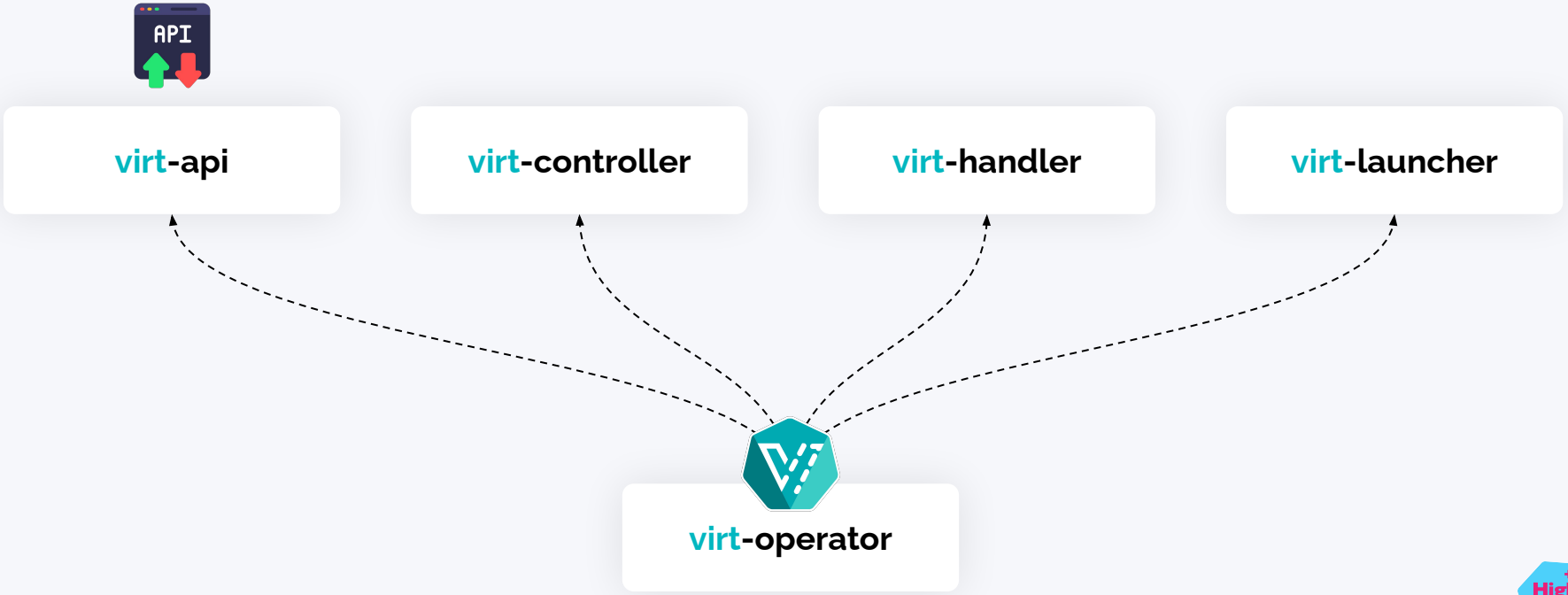
virt-operator

What is KubeVirt?

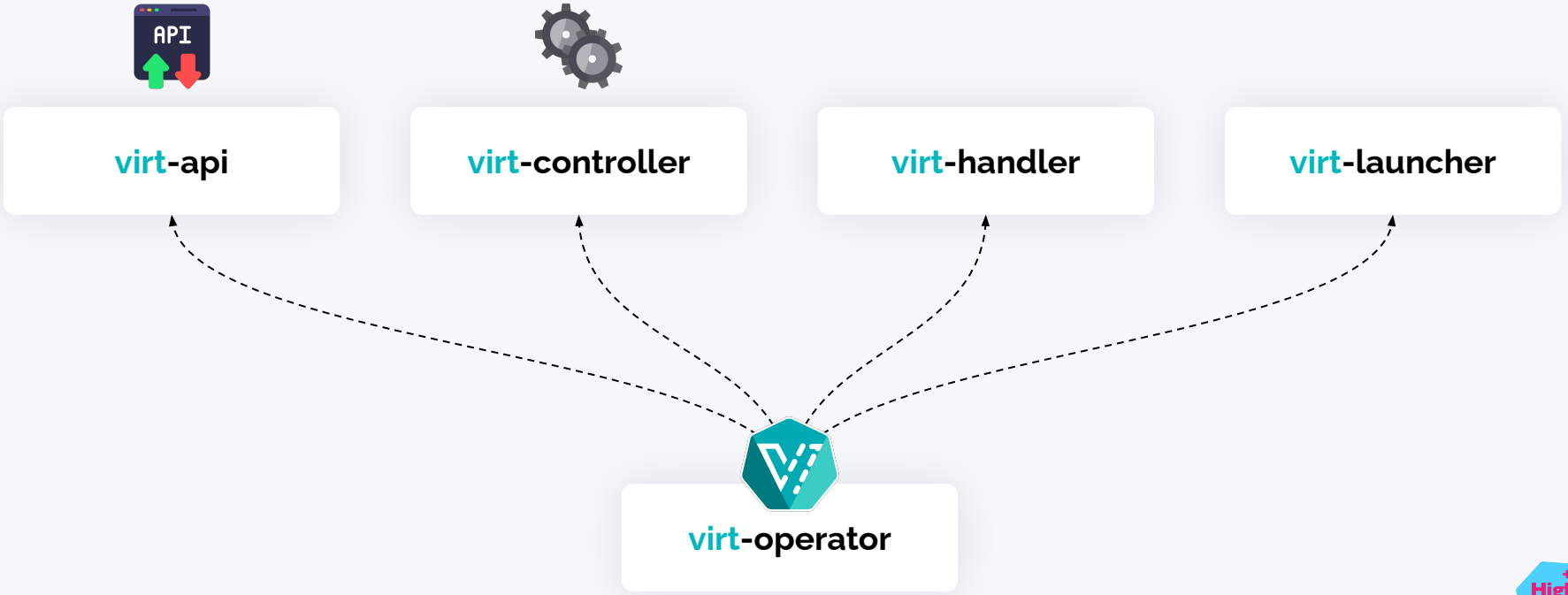
What is KubeVirt?



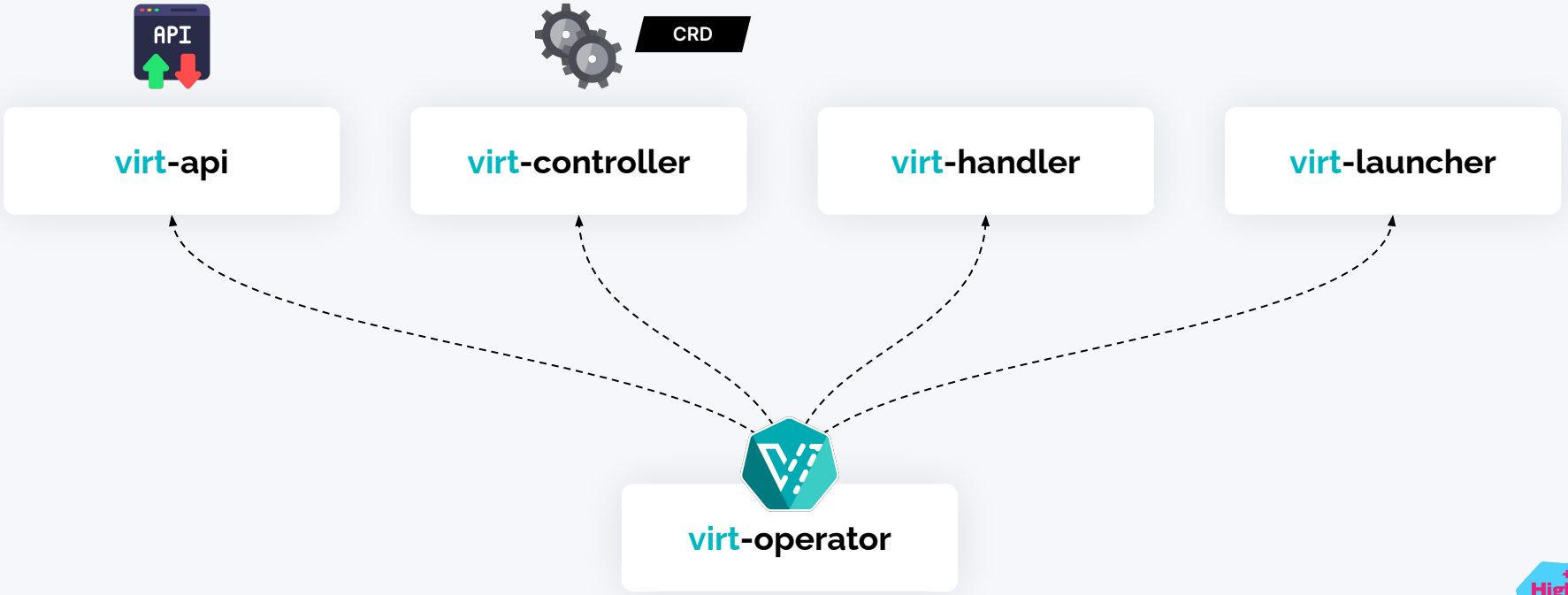
What is KubeVirt?



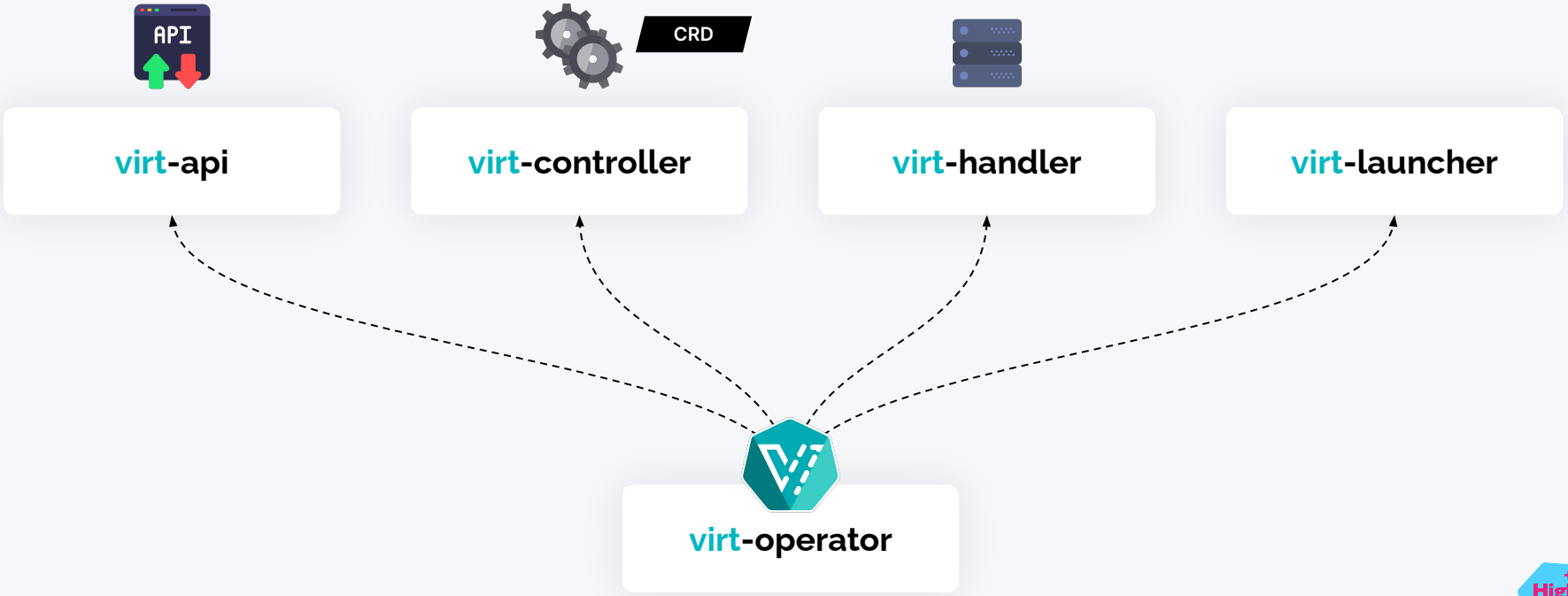
What is KubeVirt?



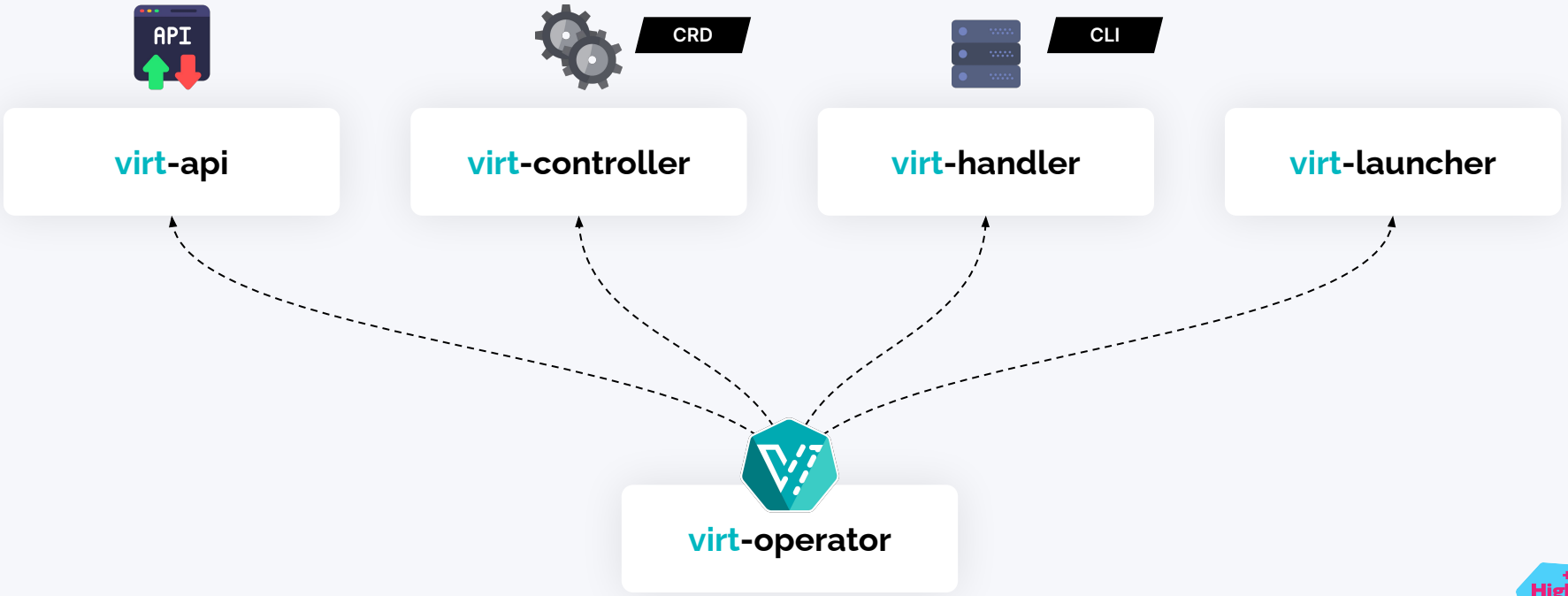
What is KubeVirt?



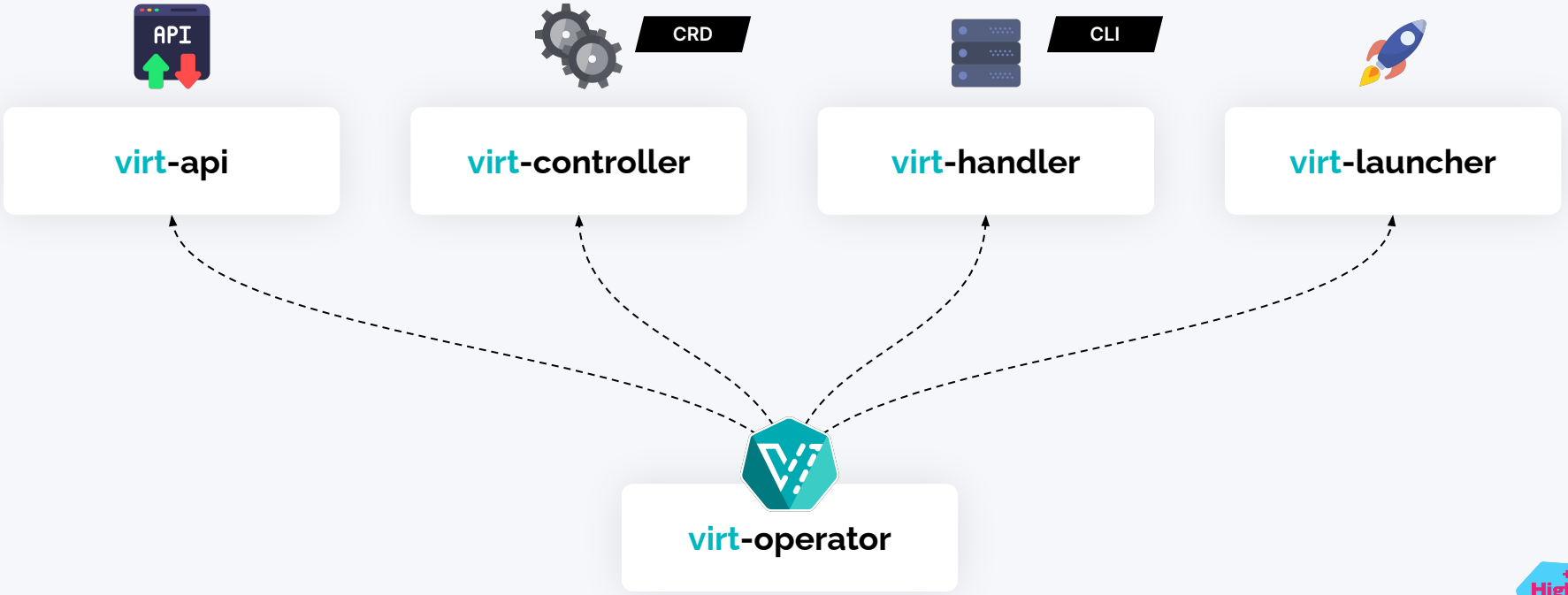
What is KubeVirt?



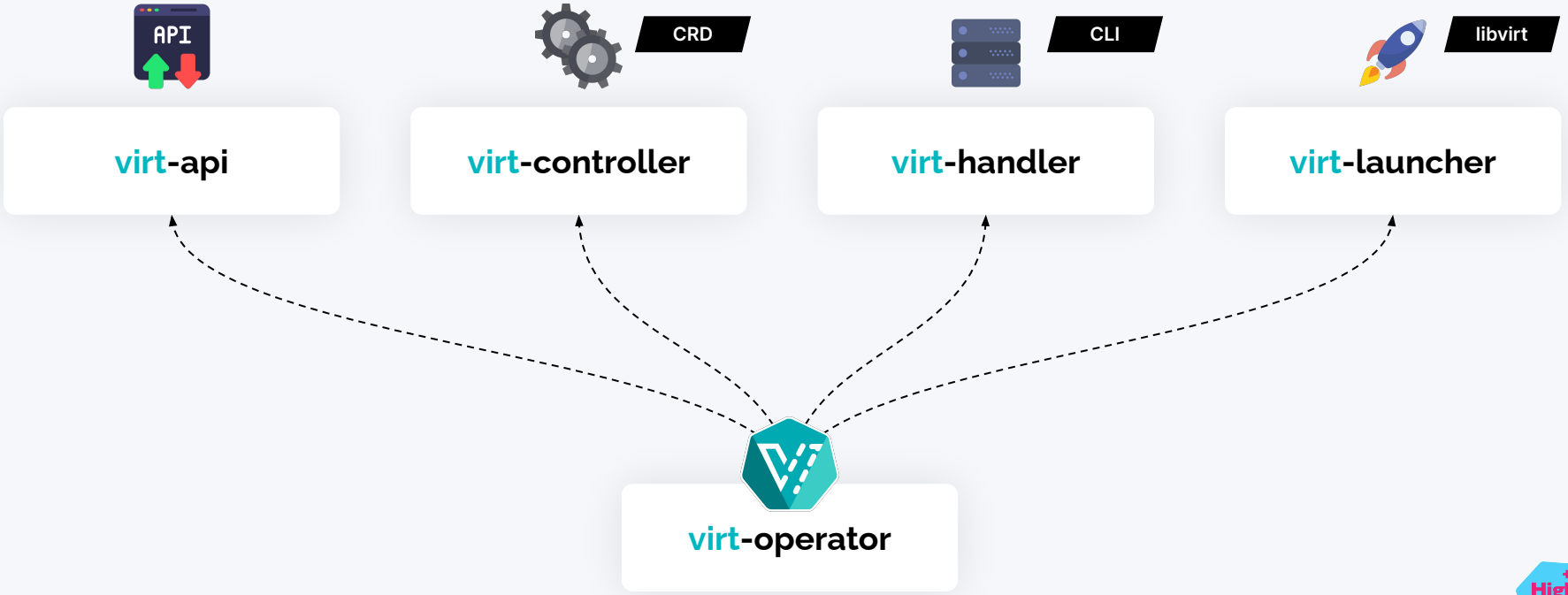
What is KubeVirt?



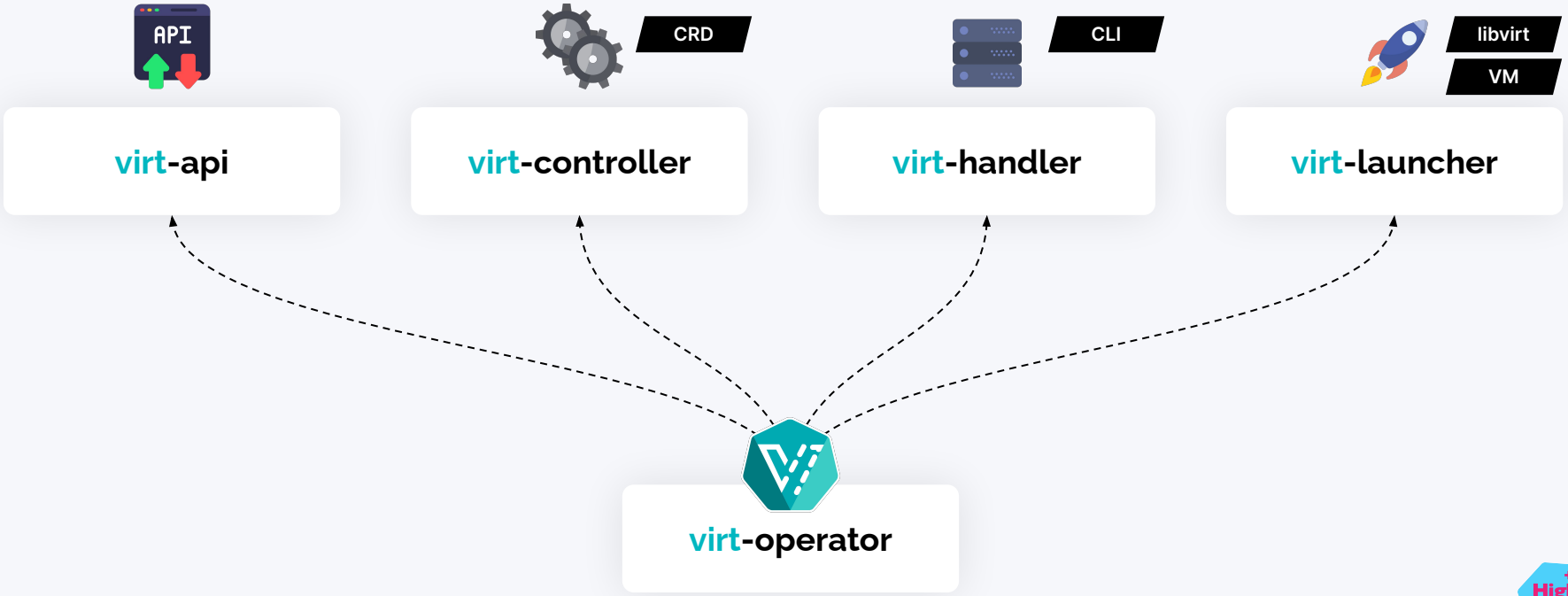
What is KubeVirt?



What is KubeVirt?



What is KubeVirt?



How does this work?

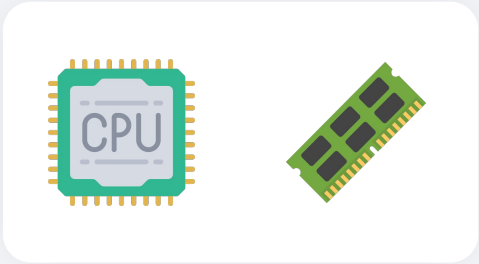
How does this work?

VM

How does this work?

VM

1



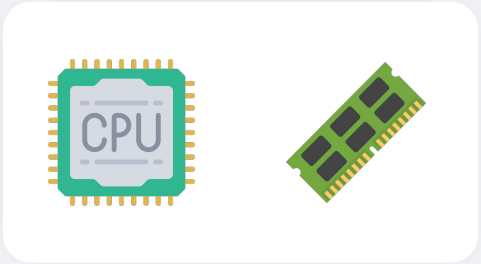
Kubernetes Runtime

CRI

How does this work?

VM

1



Kubernetes Runtime

CRI

2

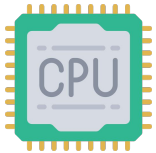
Kubernetes Storage

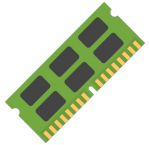
CSI

How does this work?

VM

1





Kubernetes Runtime

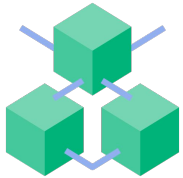
CRI

2

Kubernetes Storage

CSI

3



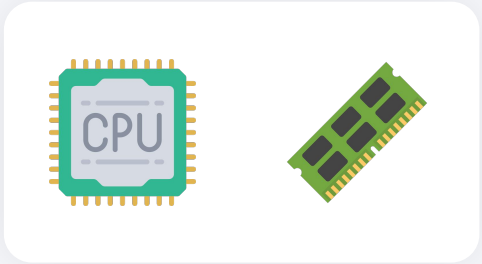
Kubernetes networking

CNI

How does this work?

VM

1



Kubernetes Runtime

CRI

2

Kubernetes Storage

CSI

3

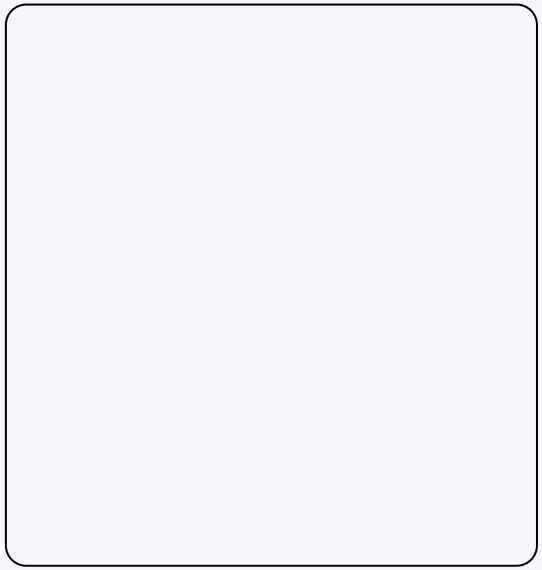
Kubernetes networking

CNI

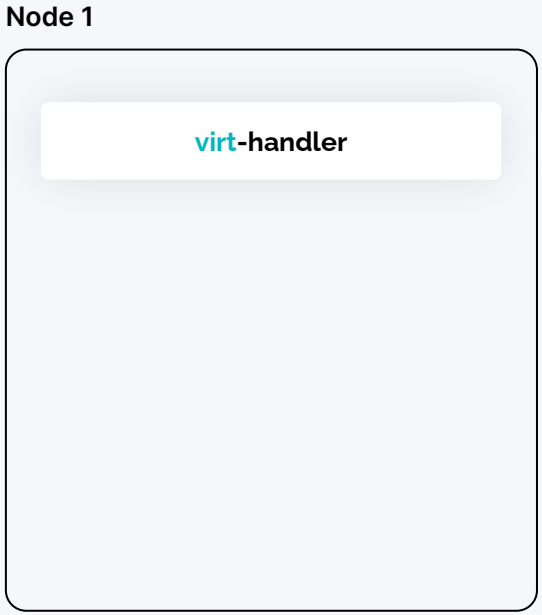
Kubernetes Runtime

Kubernetes Runtime

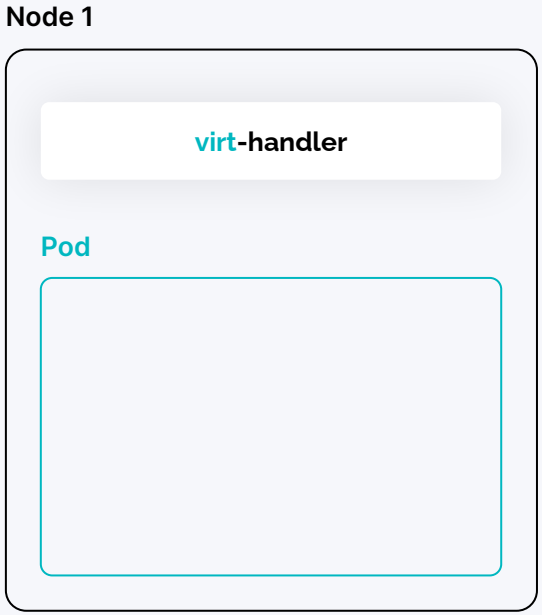
Node 1



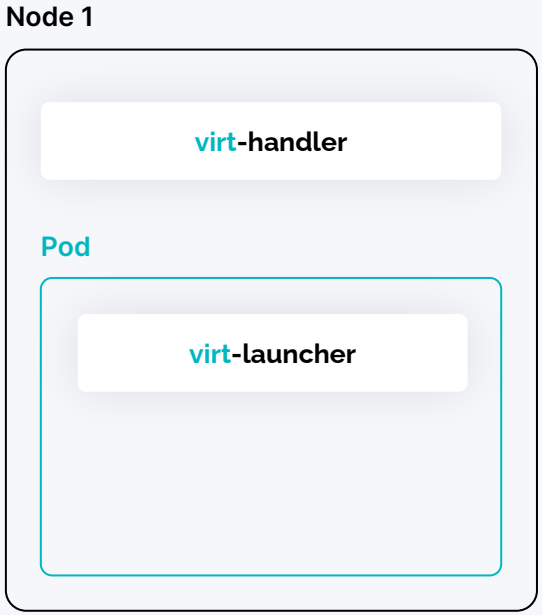
Kubernetes Runtime



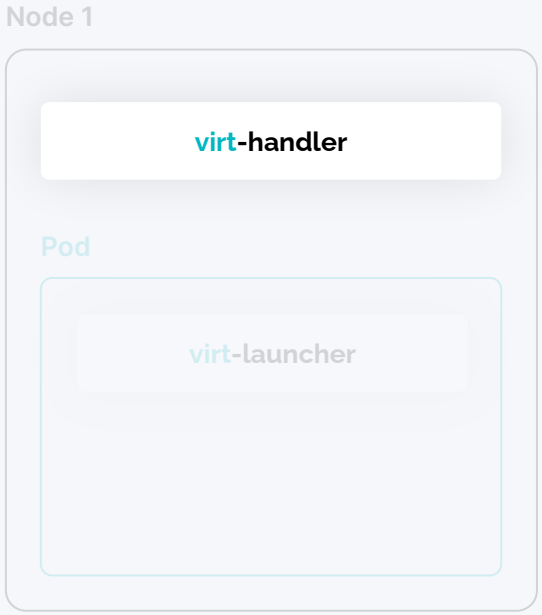
Kubernetes Runtime



Kubernetes Runtime

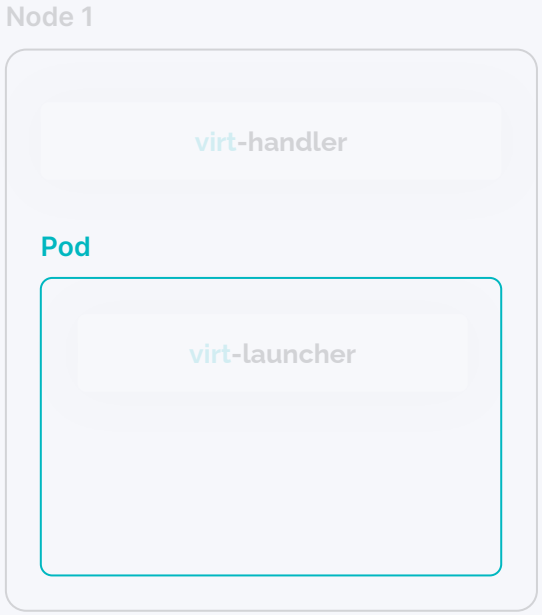


Kubernetes Runtime



Has privileges and access to host OS

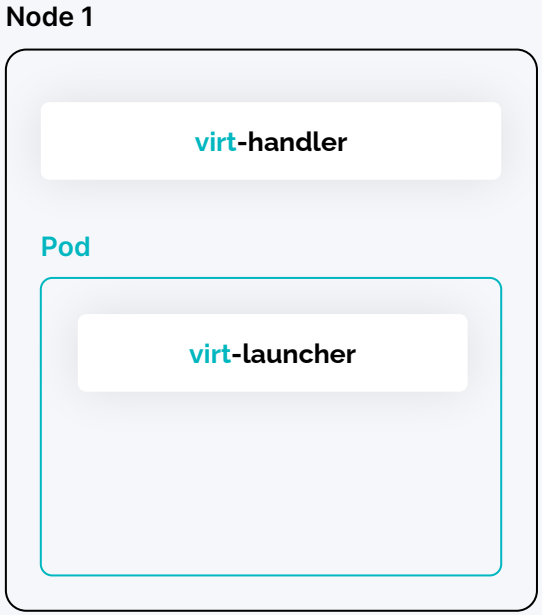
Kubernetes Runtime



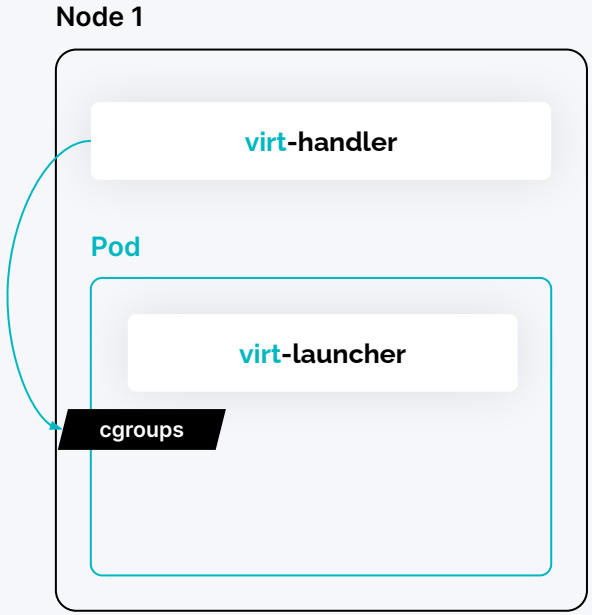
Has privileges and access to host OS

Unprivileged pod

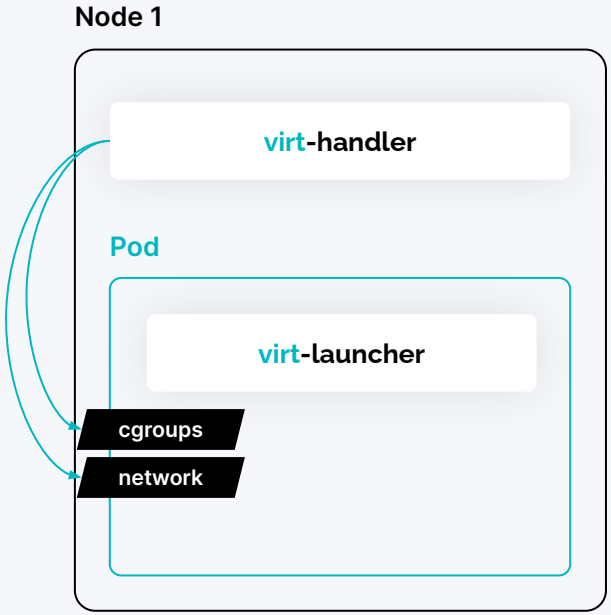
Kubernetes Runtime



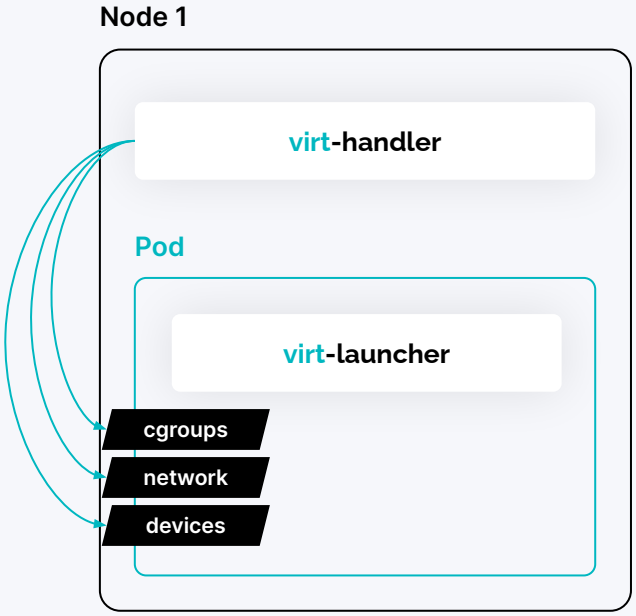
Kubernetes Runtime



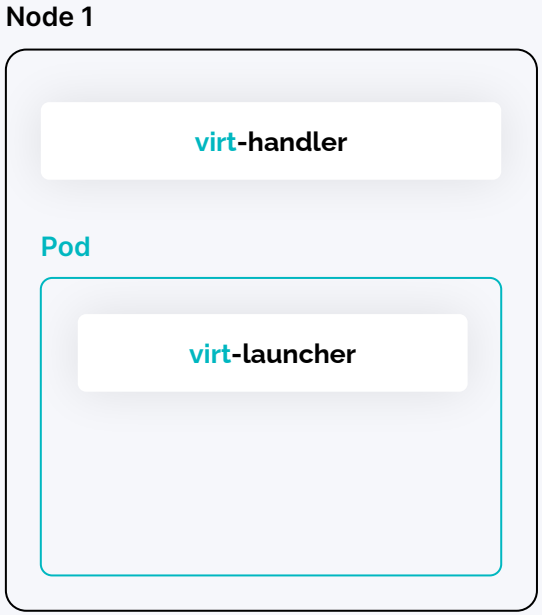
Kubernetes Runtime



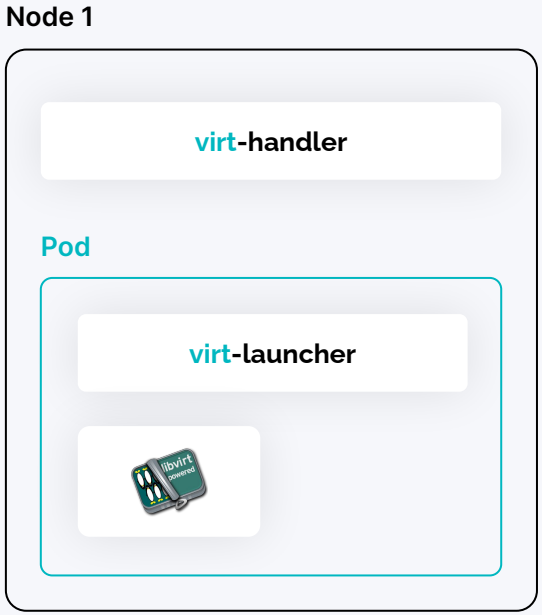
Kubernetes Runtime



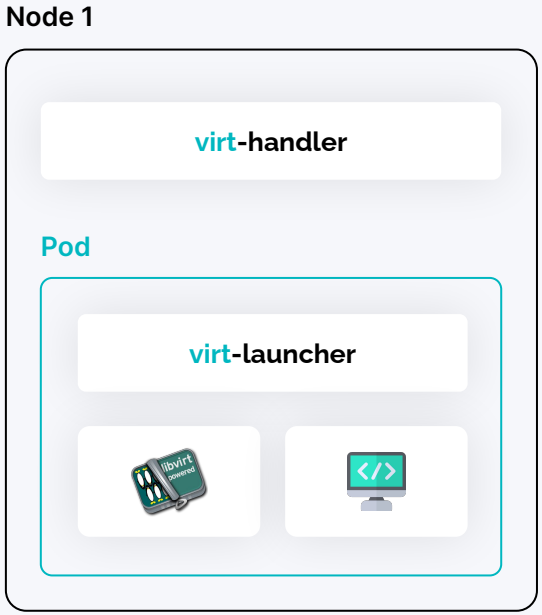
Kubernetes Runtime



Kubernetes Runtime



Kubernetes Runtime



Kubernetes Runtime: Live migration

Node 1

virt-handler

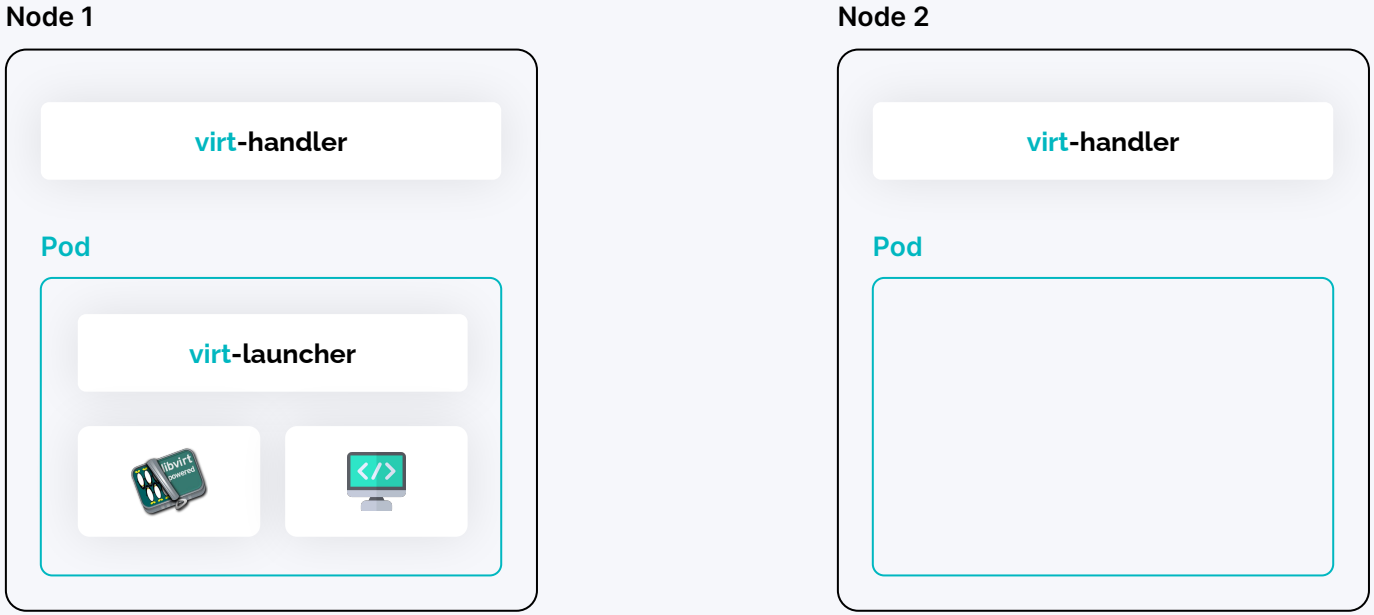
Pod

virt-launcher

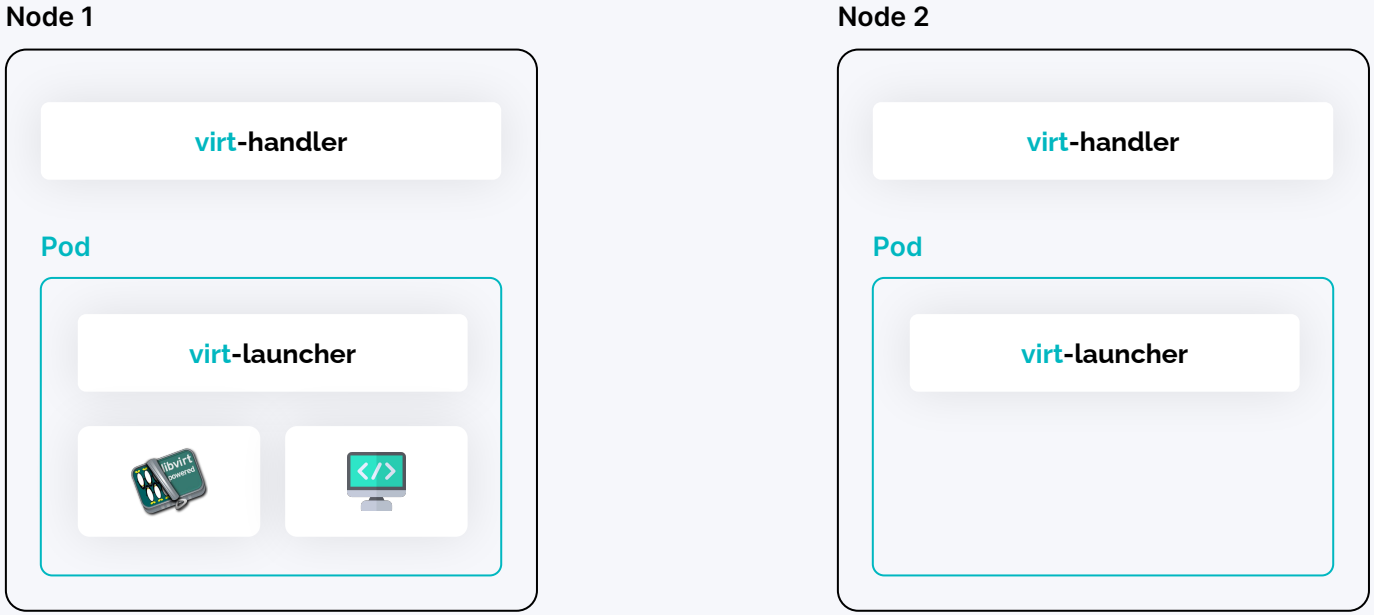
Node 2

virt-handler

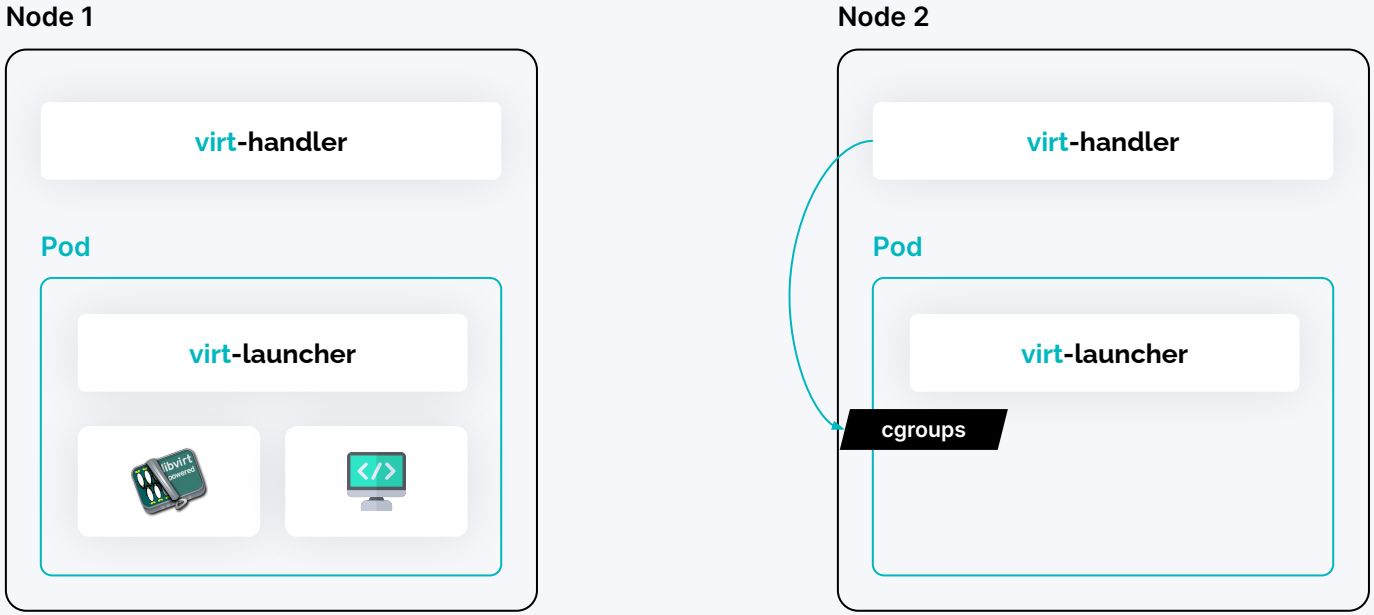
Kubernetes Runtime: Live migration



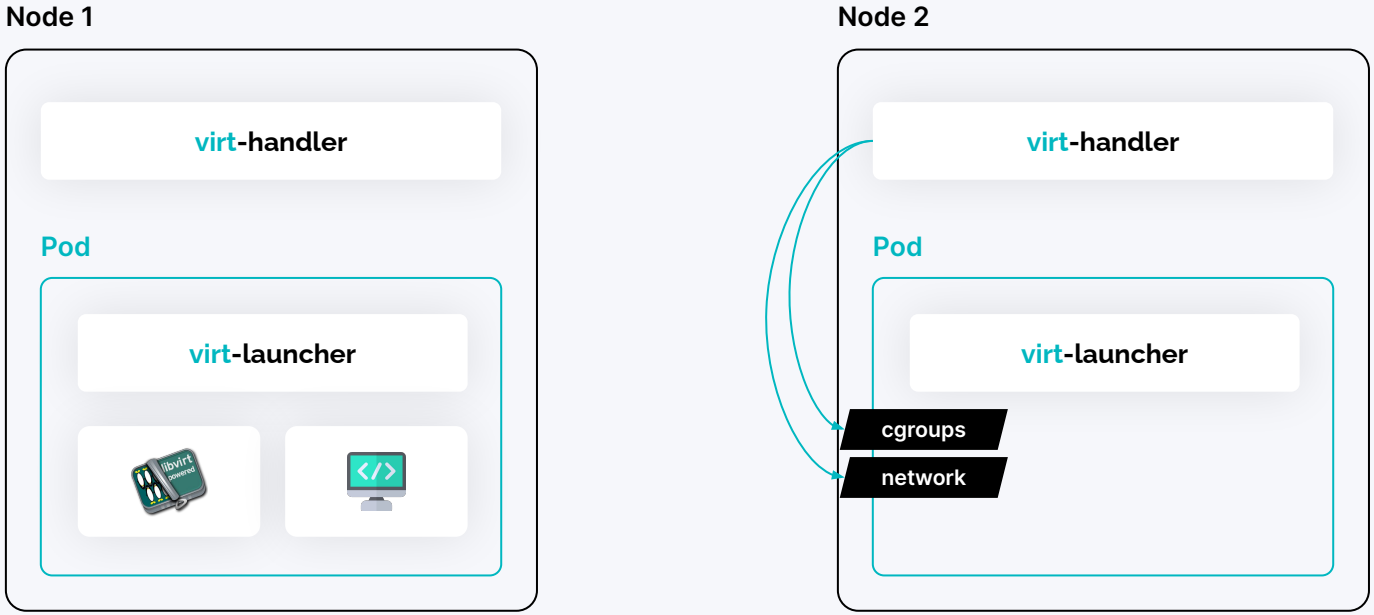
Kubernetes Runtime: Live migration



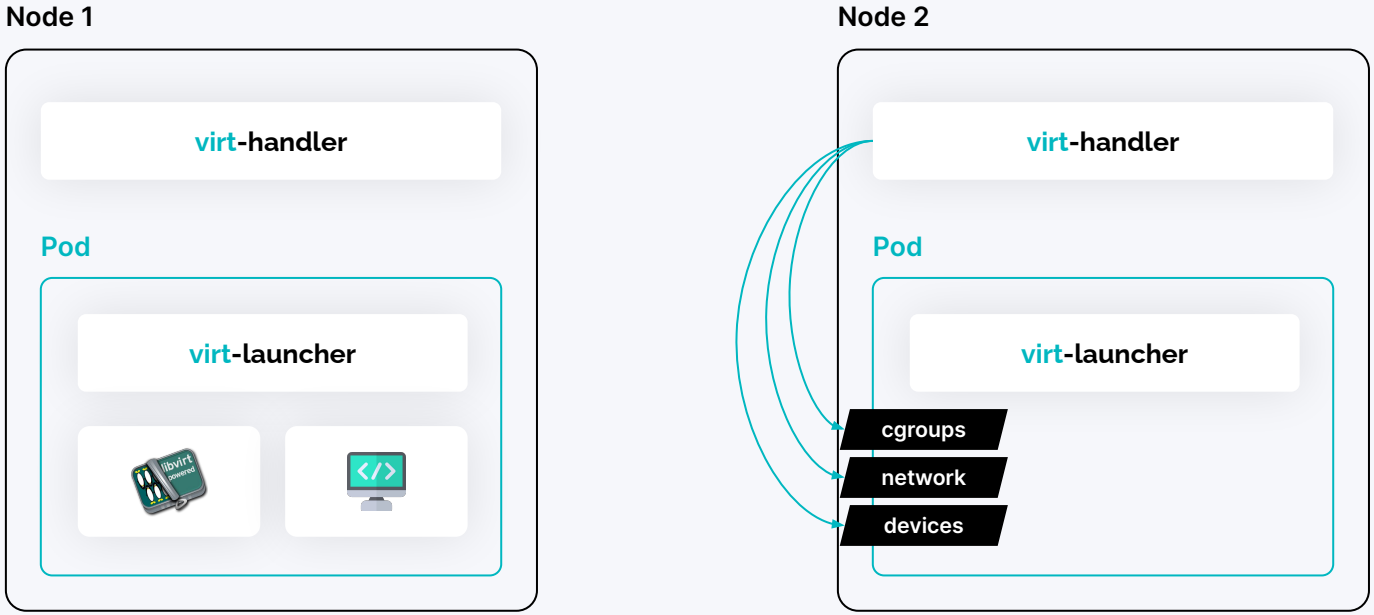
Kubernetes Runtime: Live migration



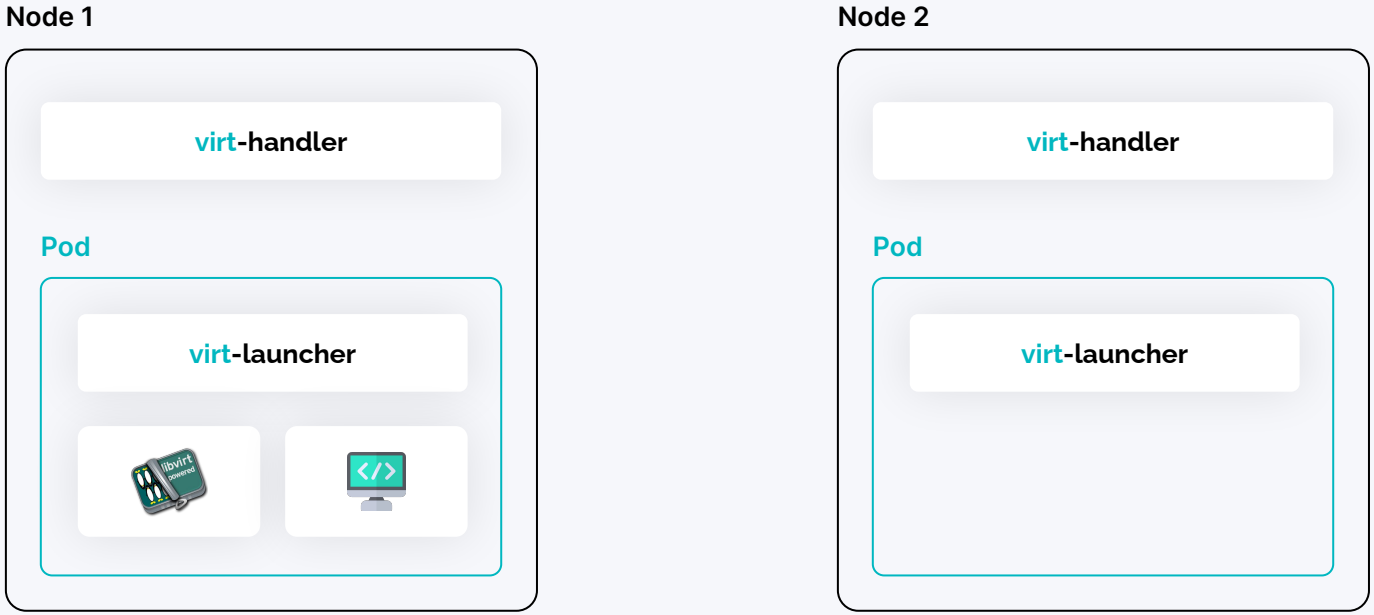
Kubernetes Runtime: Live migration



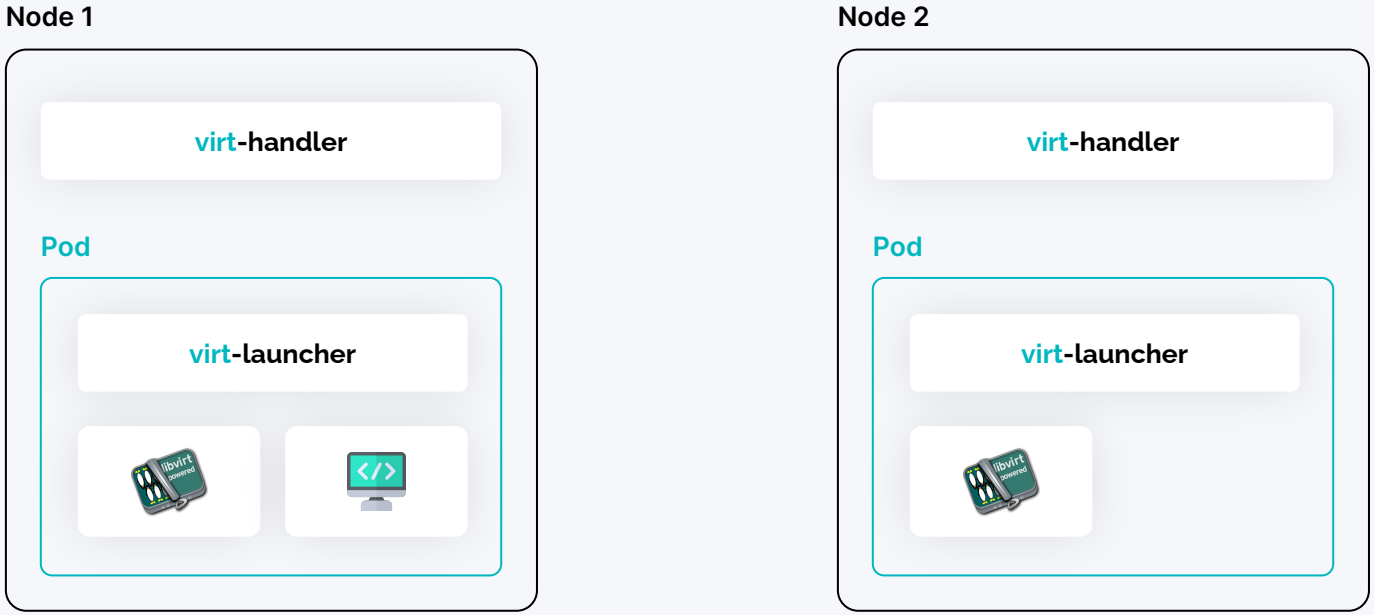
Kubernetes Runtime: Live migration



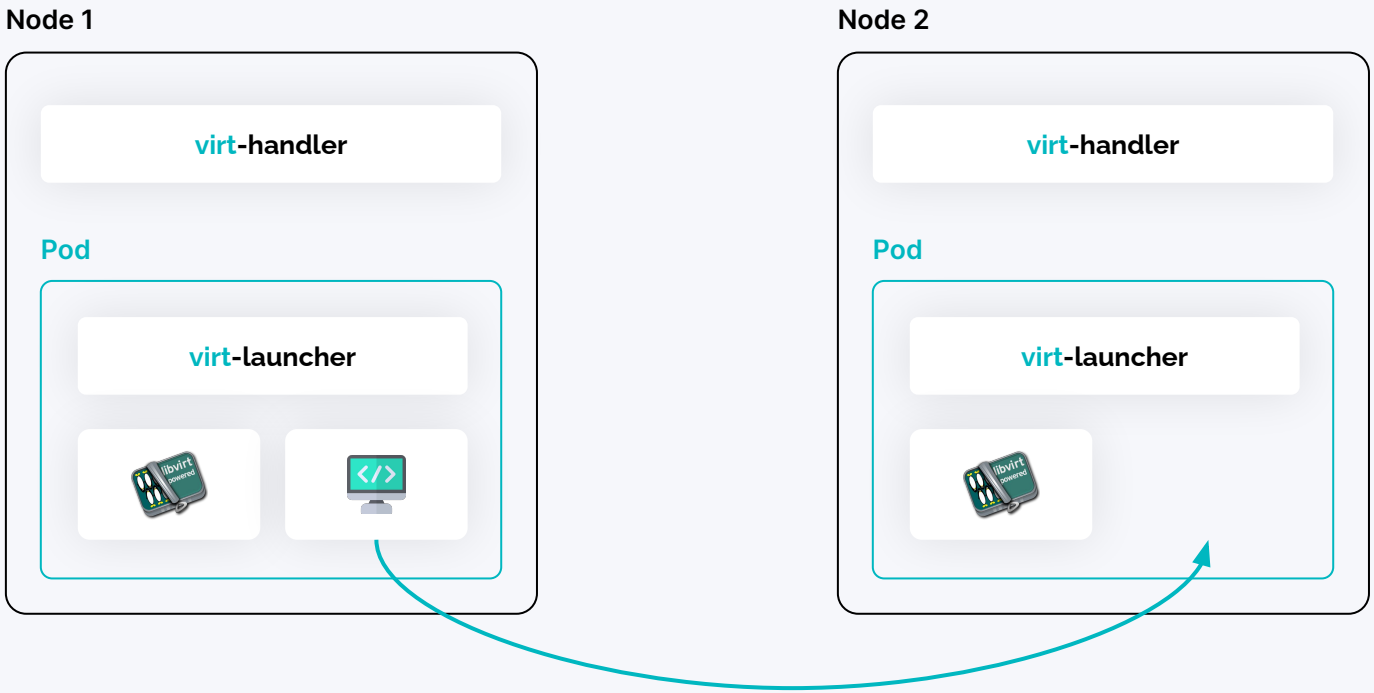
Kubernetes Runtime: Live migration



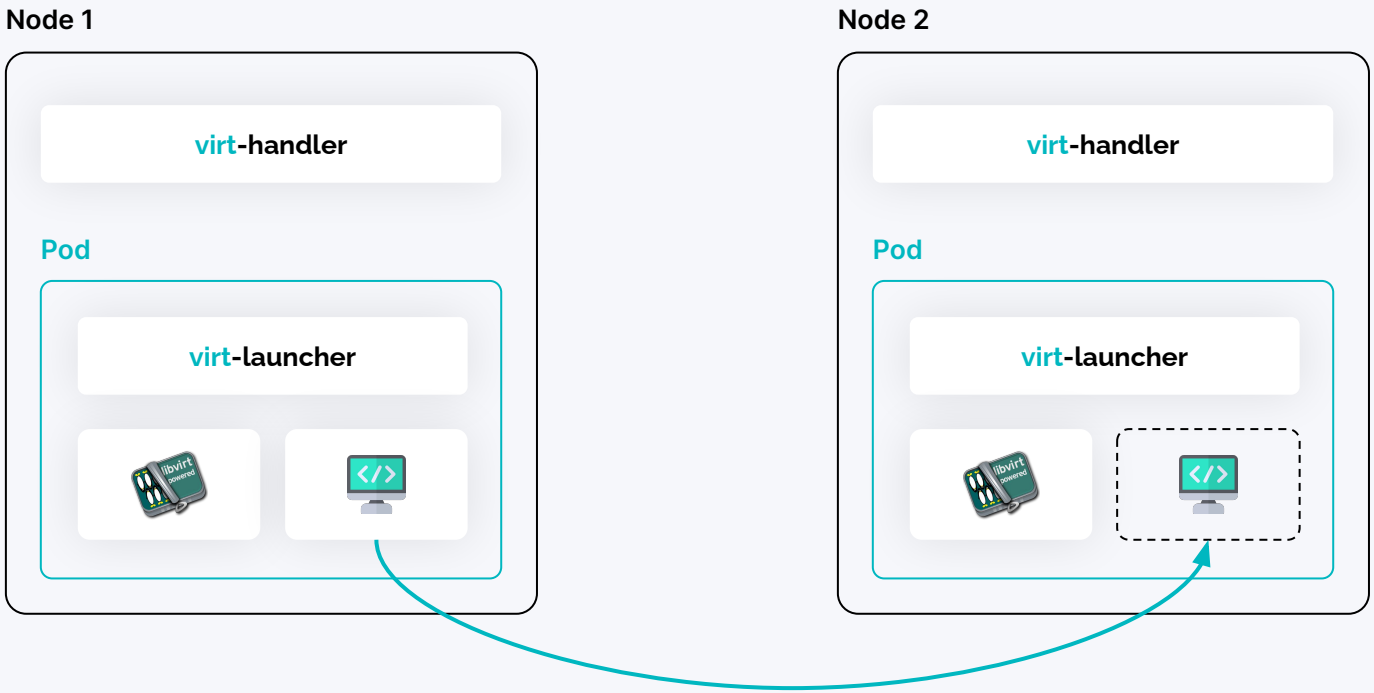
Kubernetes Runtime: Live migration



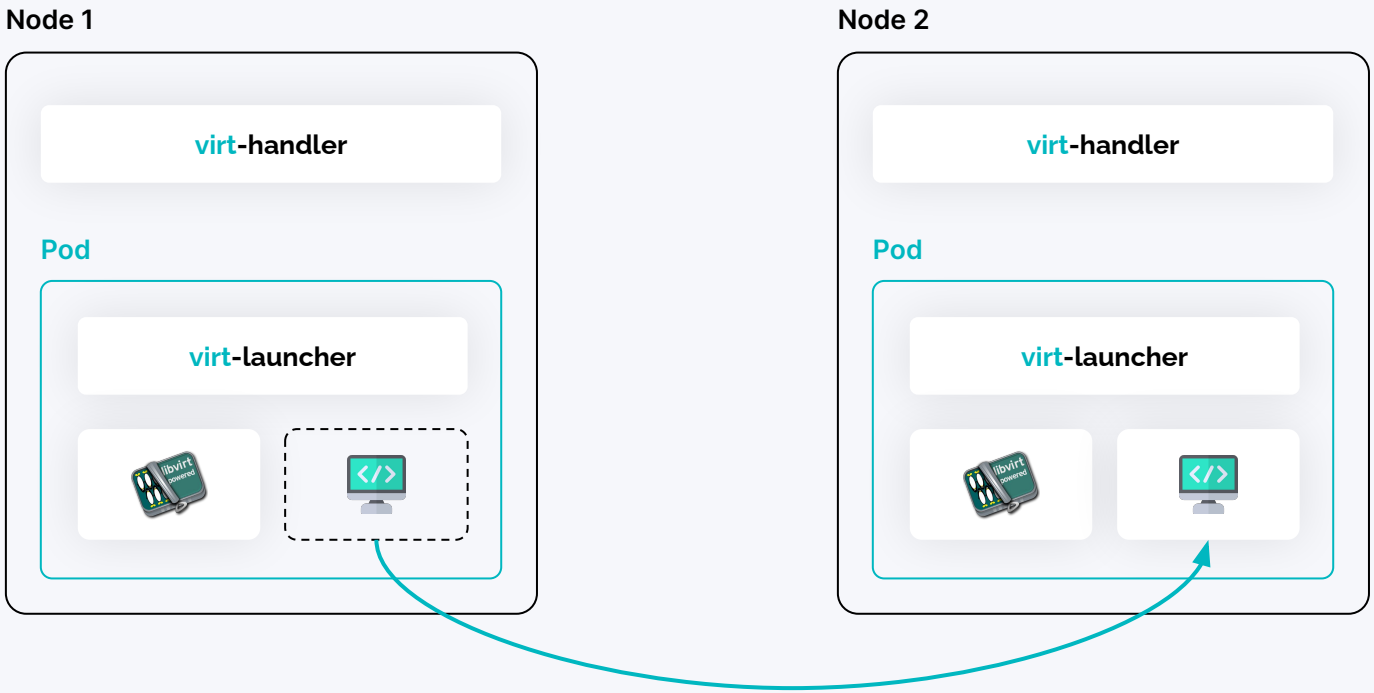
Kubernetes Runtime: Live migration



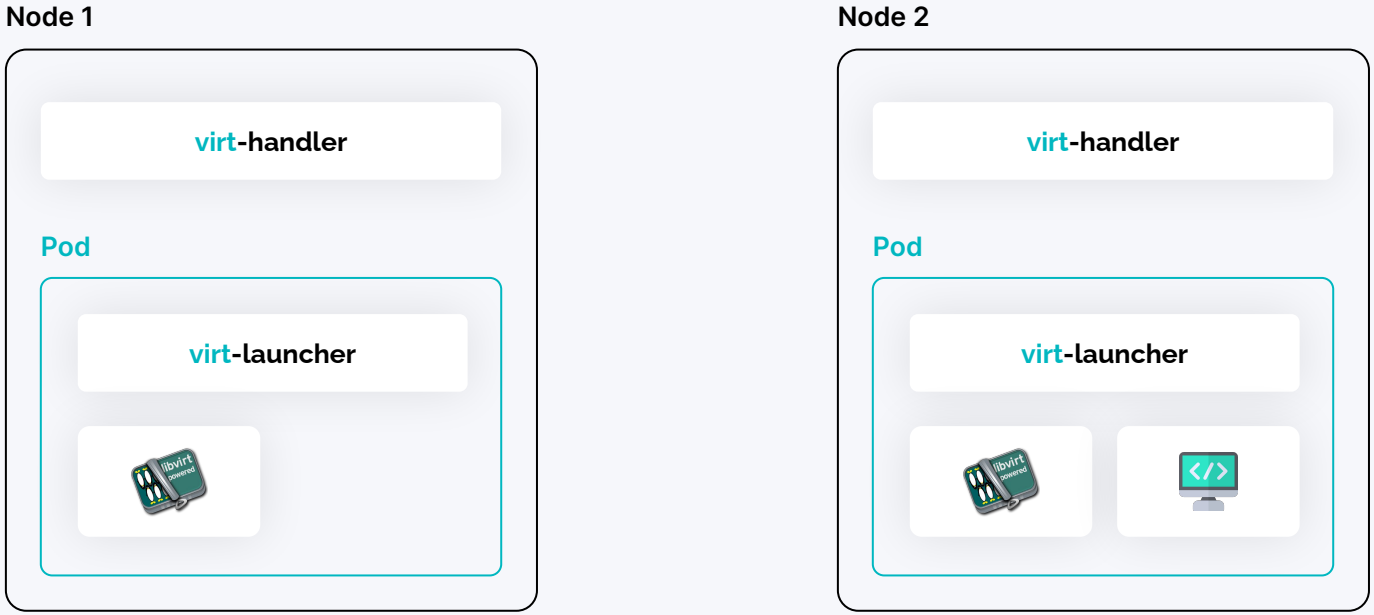
Kubernetes Runtime: Live migration



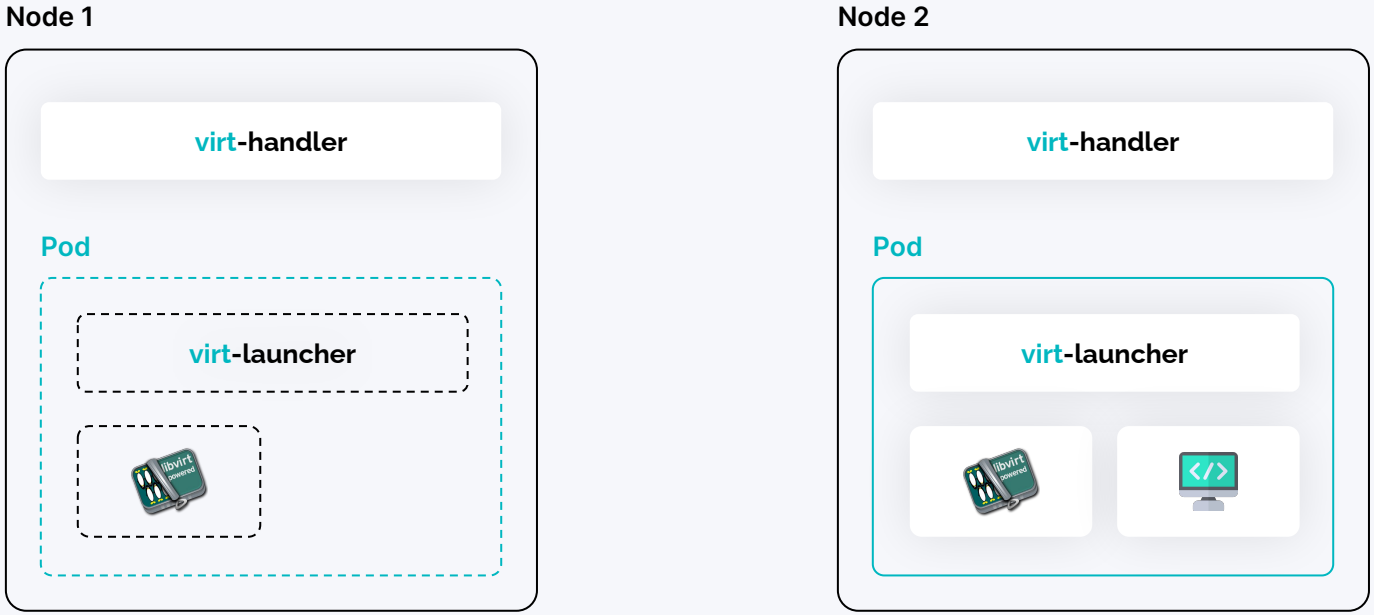
Kubernetes Runtime: Live migration



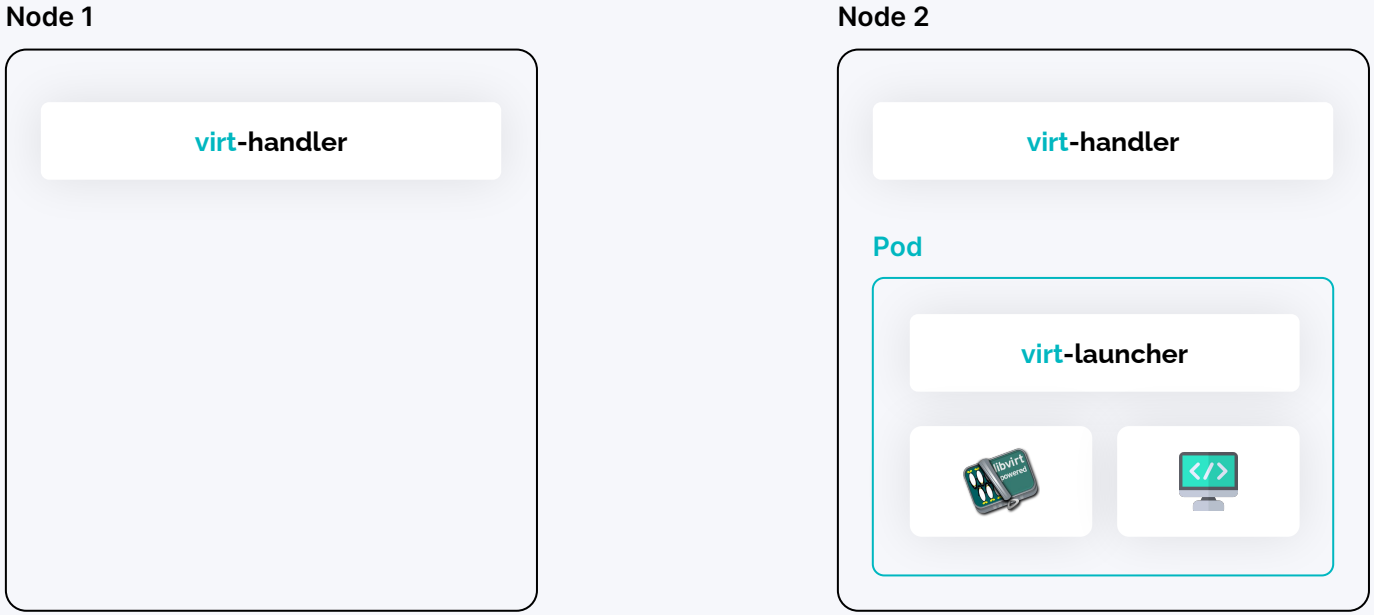
Kubernetes Runtime: Live migration



Kubernetes Runtime: Live migration



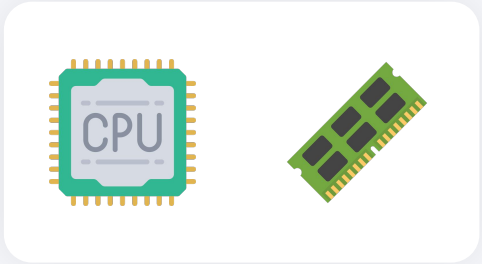
Kubernetes Runtime: Live migration



How does this work?

VM

1



Kubernetes Runtime

CRI

2

Kubernetes Storage

CSI

3

Kubernetes networking

CNI

How does this work?

VM

1

Kubernetes Runtime

CRI

2

Kubernetes Storage

CSI

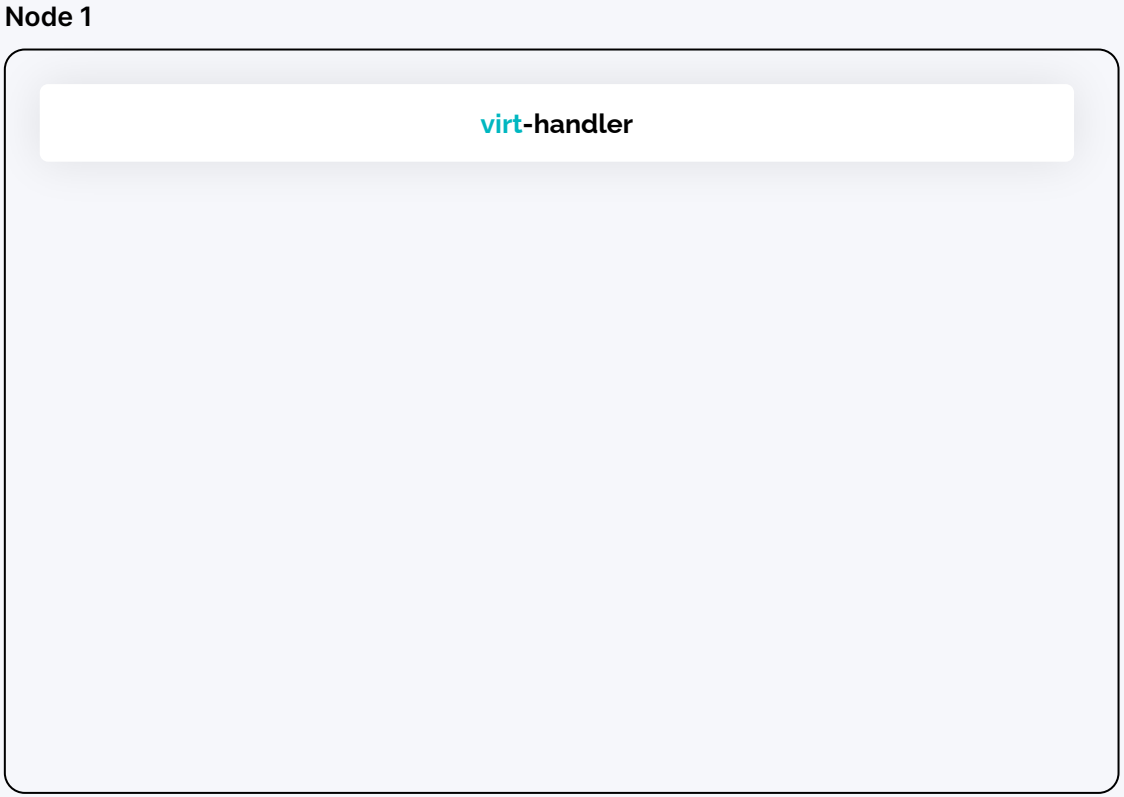
3

Kubernetes networking

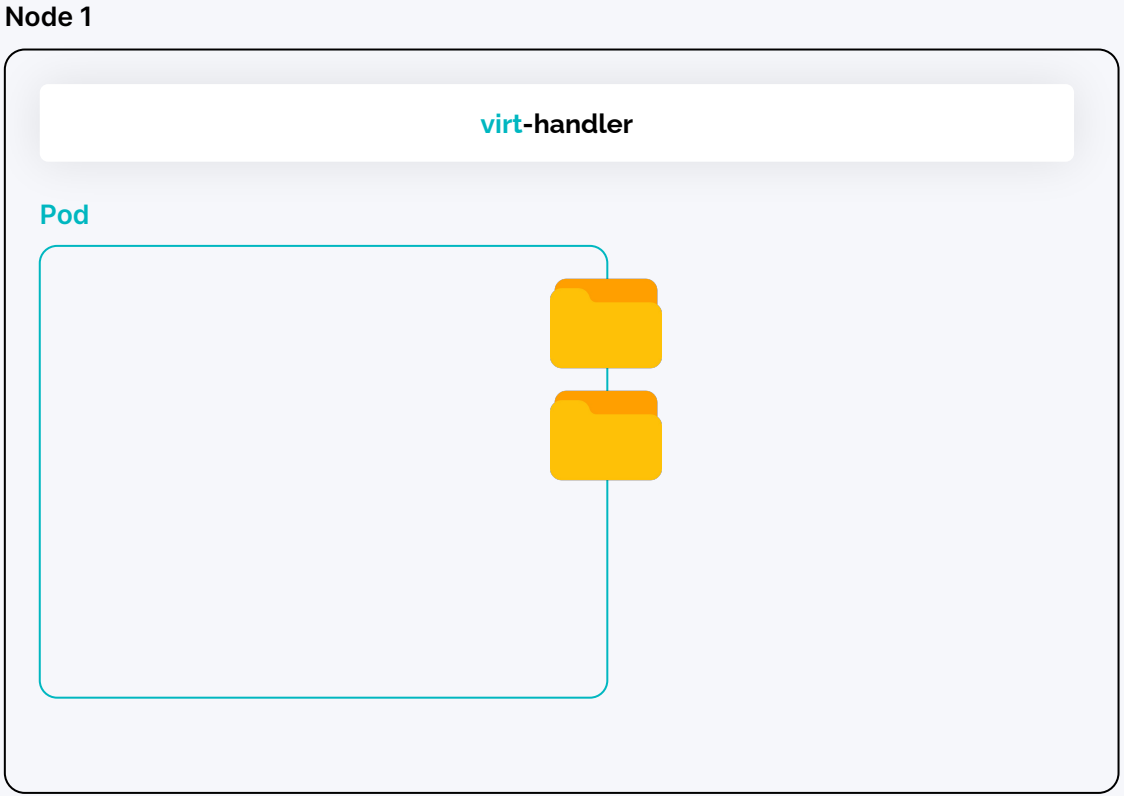
CNI

Kubernetes Storage

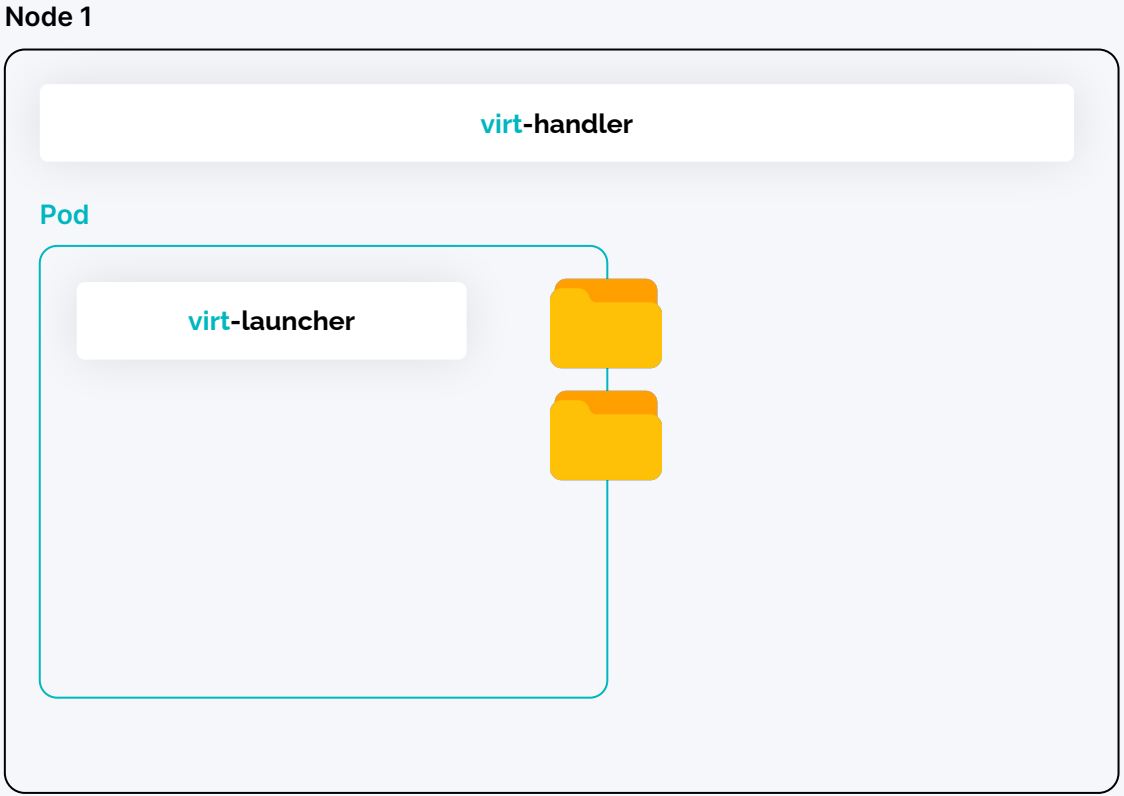
Kubernetes Storage



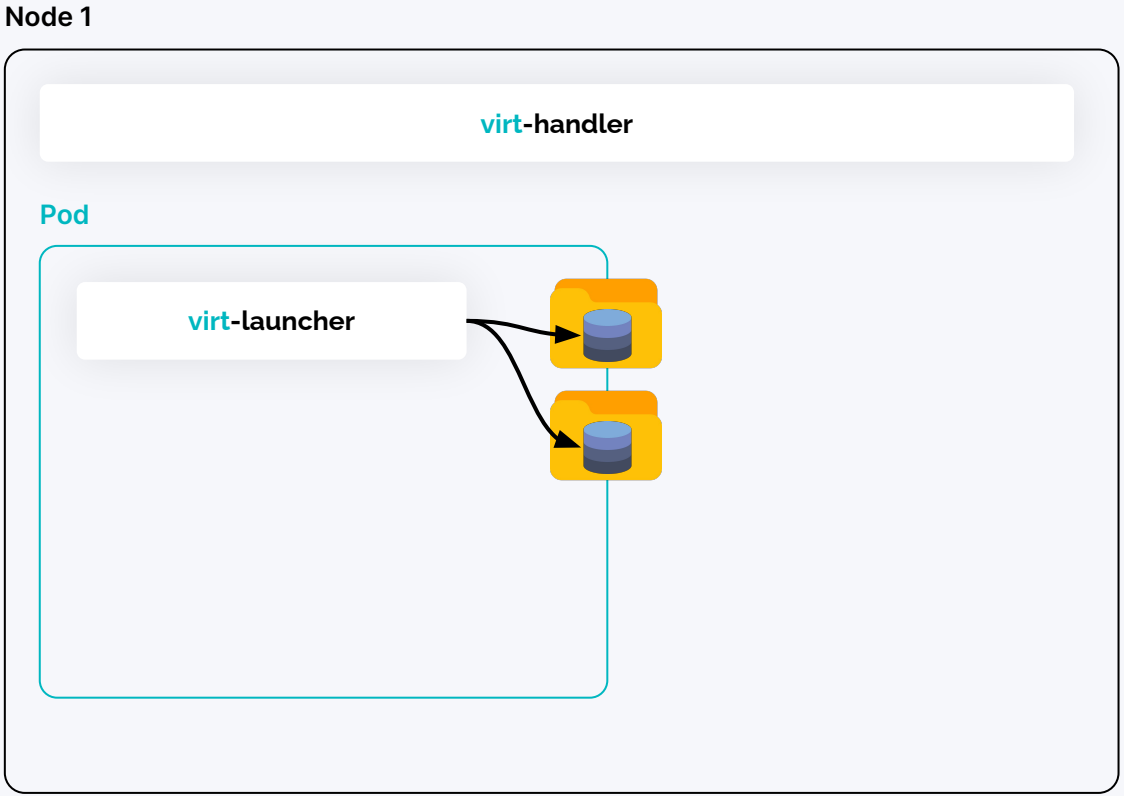
Kubernetes Storage



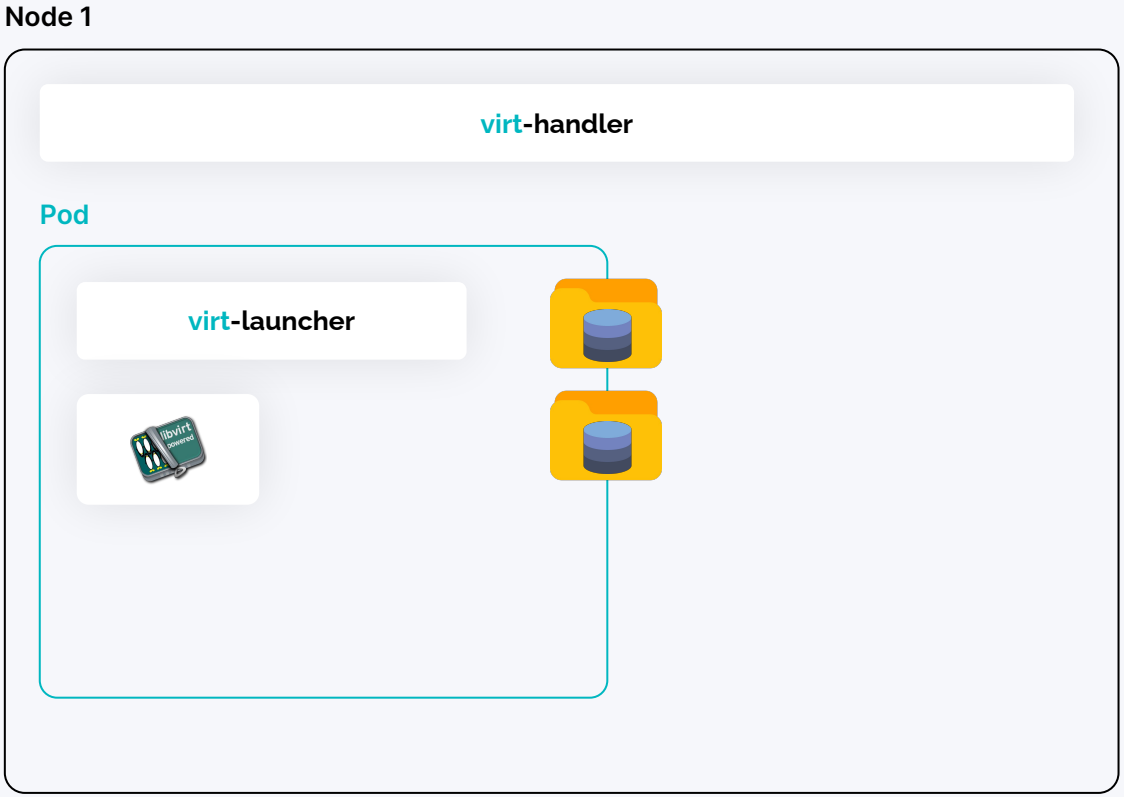
Kubernetes Storage



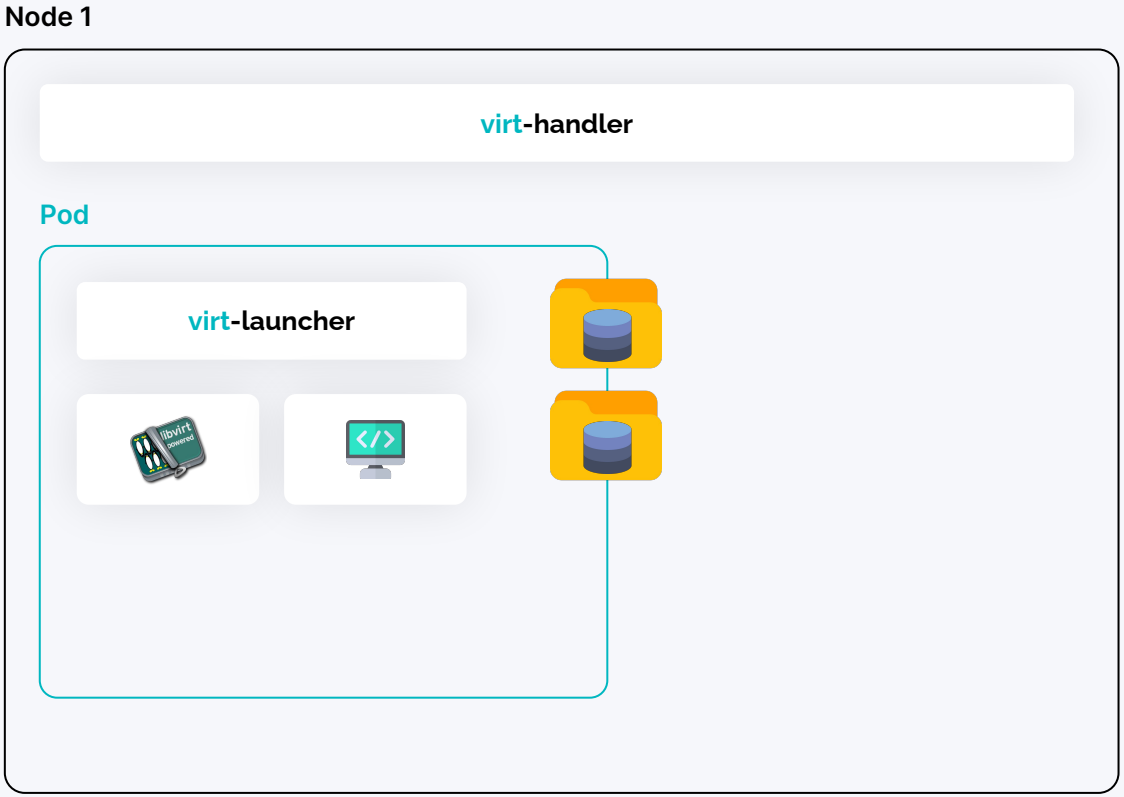
Kubernetes Storage



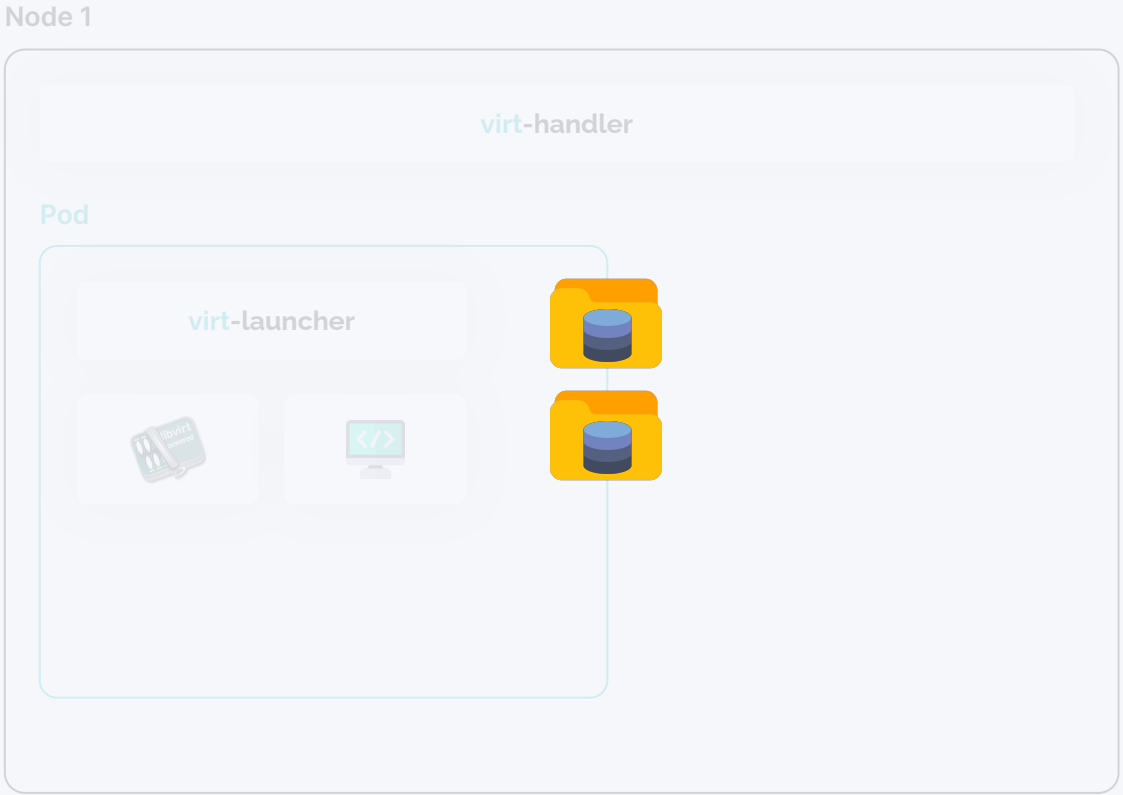
Kubernetes Storage



Kubernetes Storage

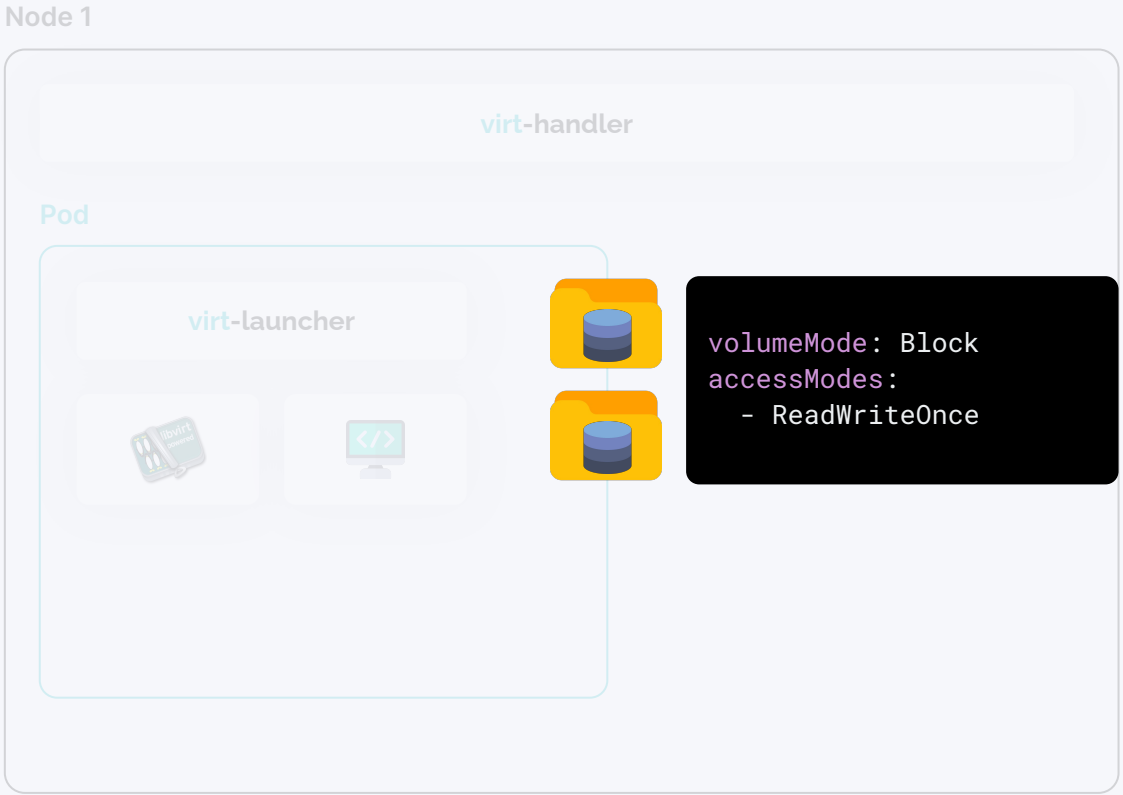


Kubernetes Storage

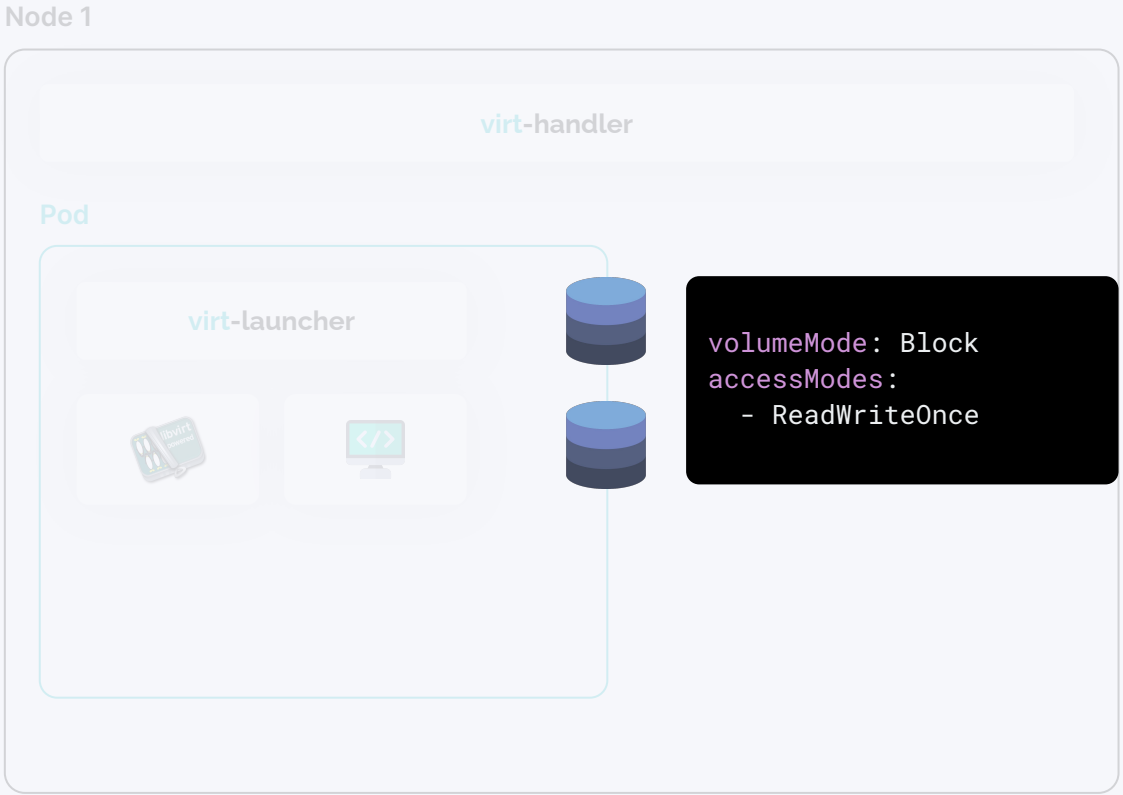


Kubernetes Storage

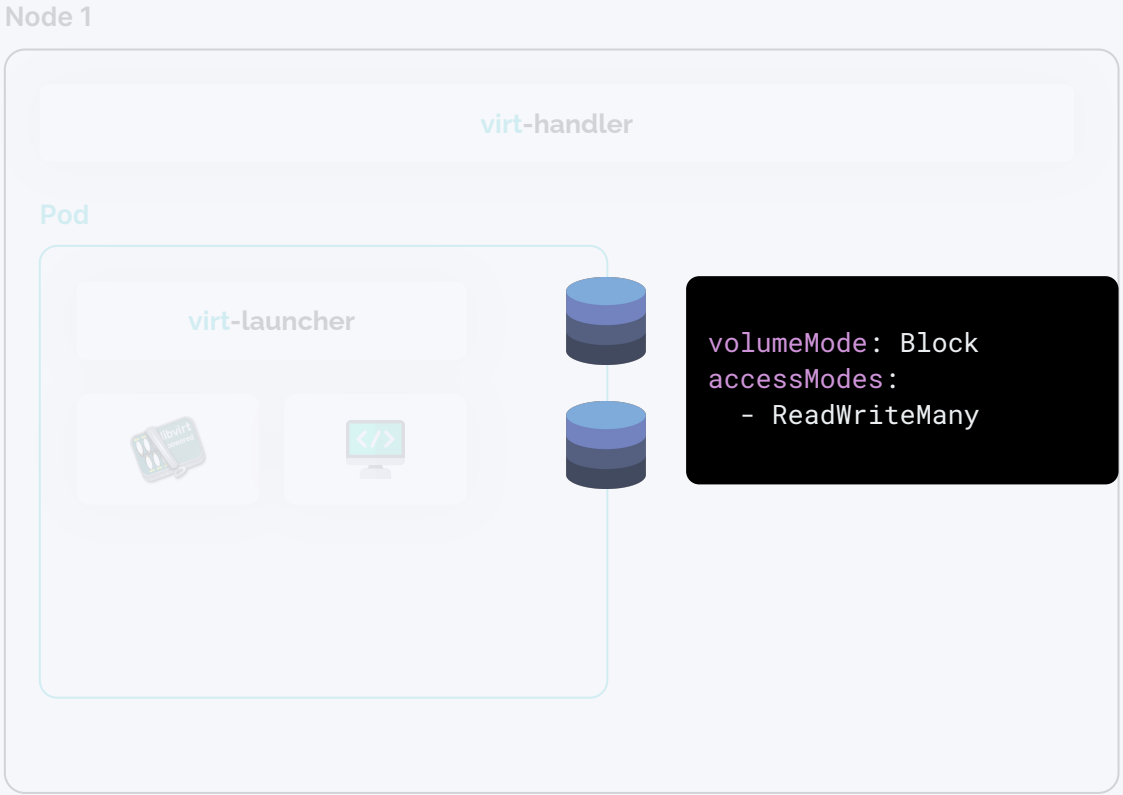
Kubernetes Storage



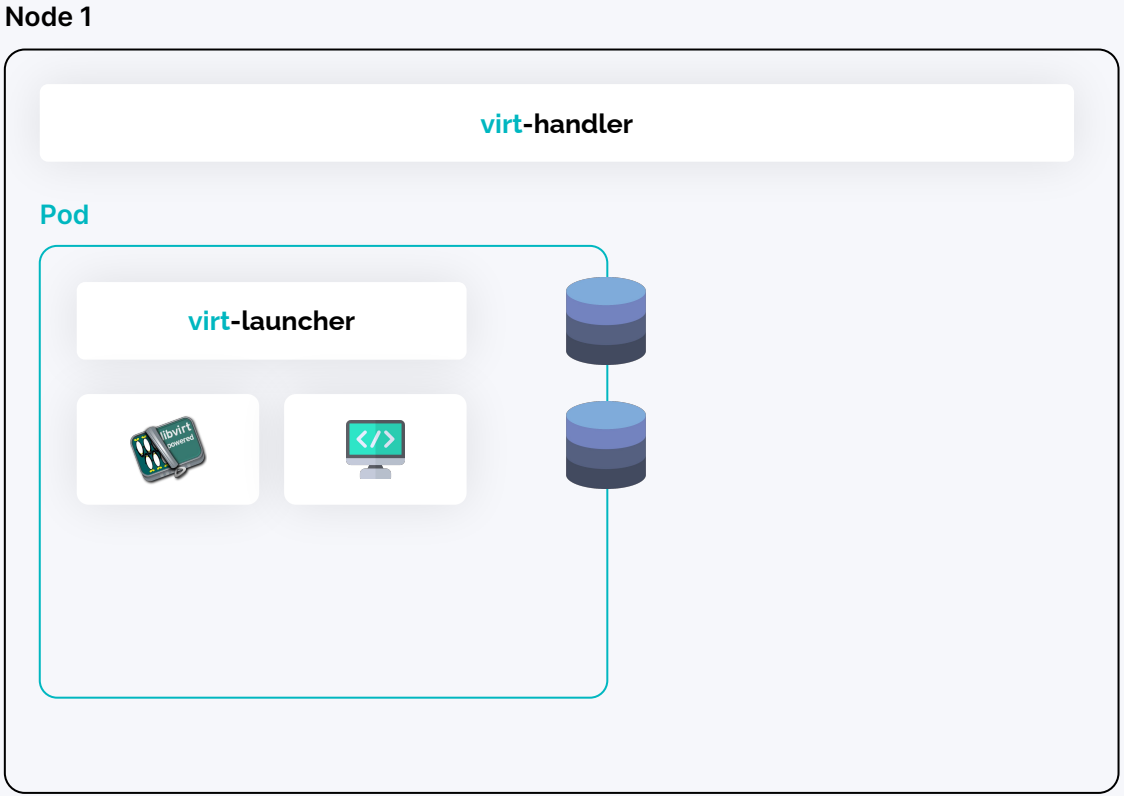
Kubernetes Storage



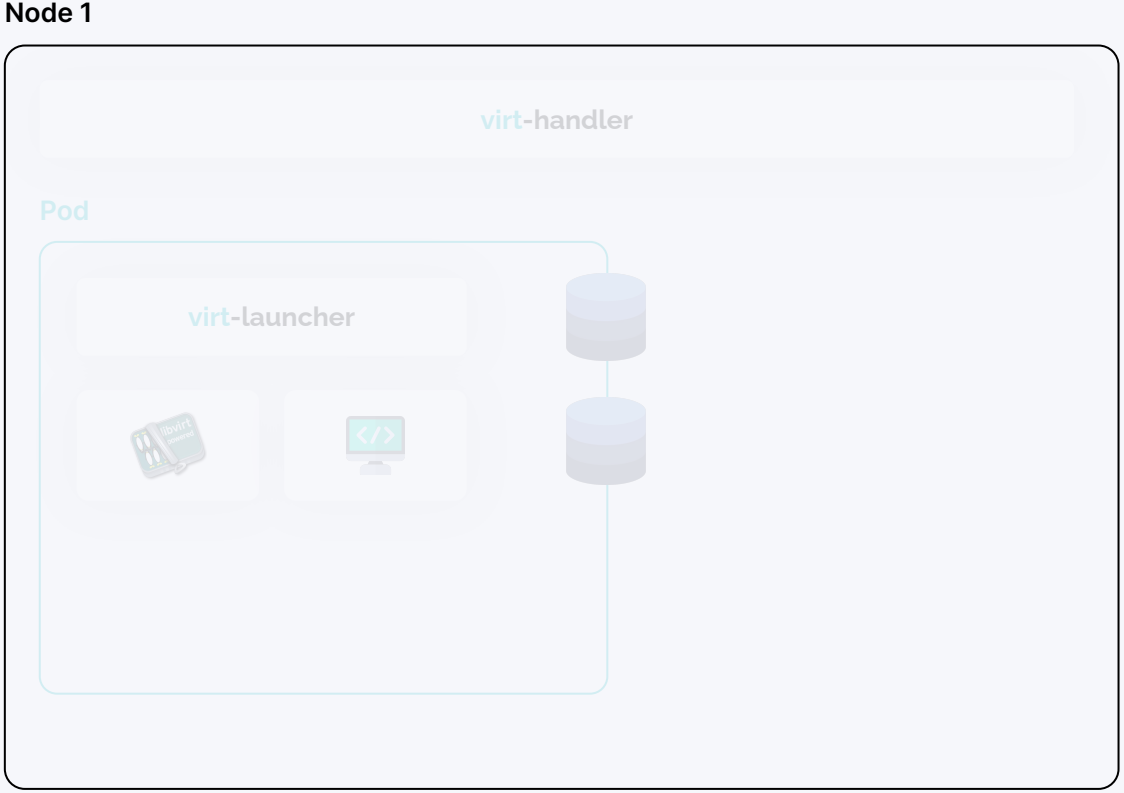
Kubernetes Storage



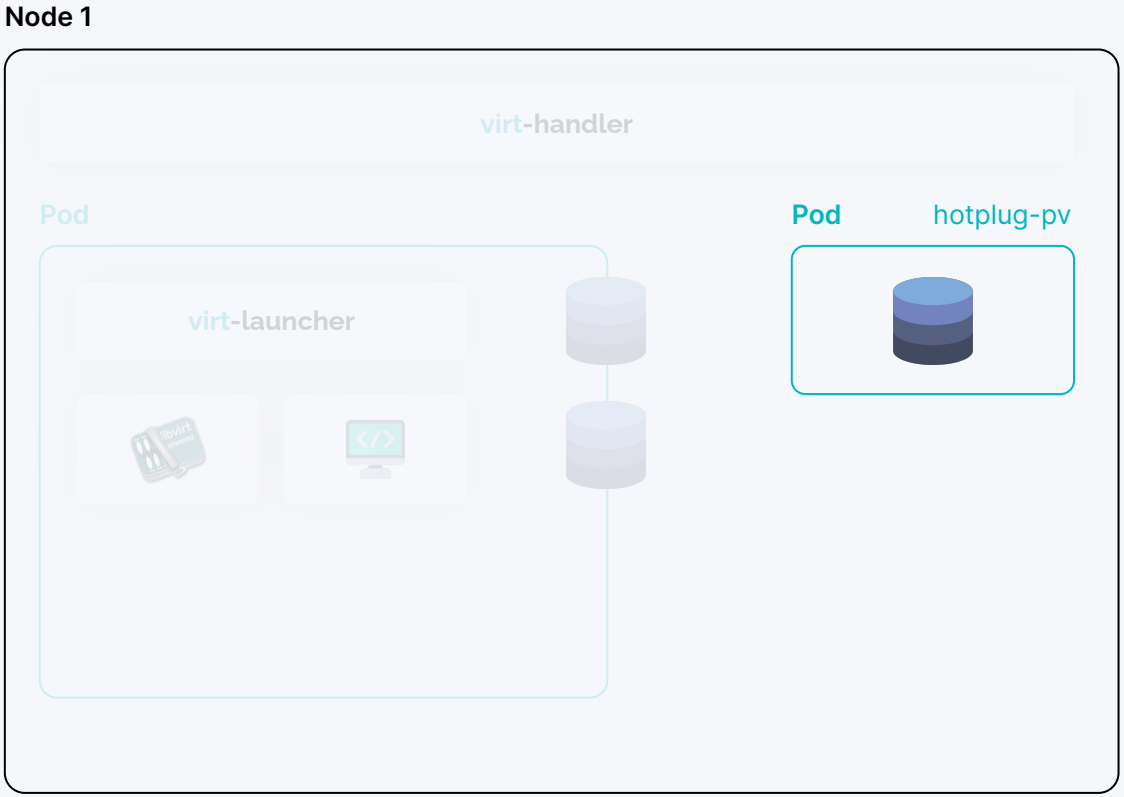
Kubernetes Storage: hotplugging volumes



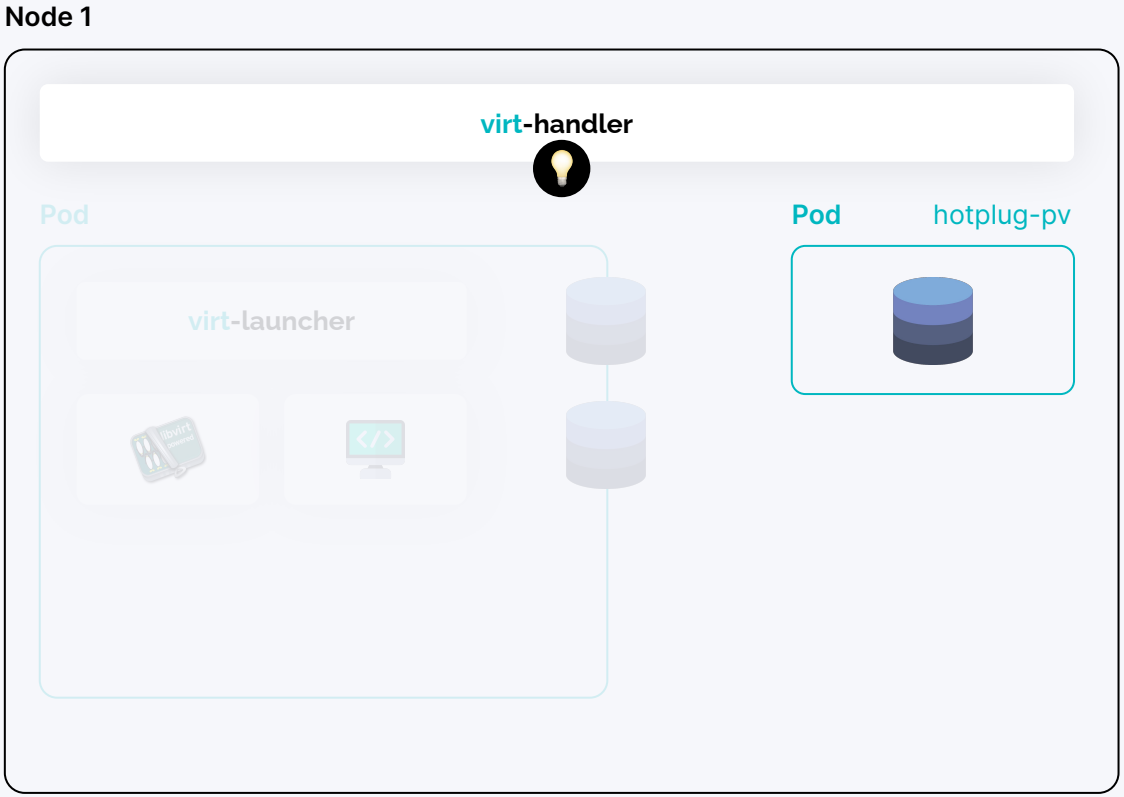
Kubernetes Storage: hotplugging volumes



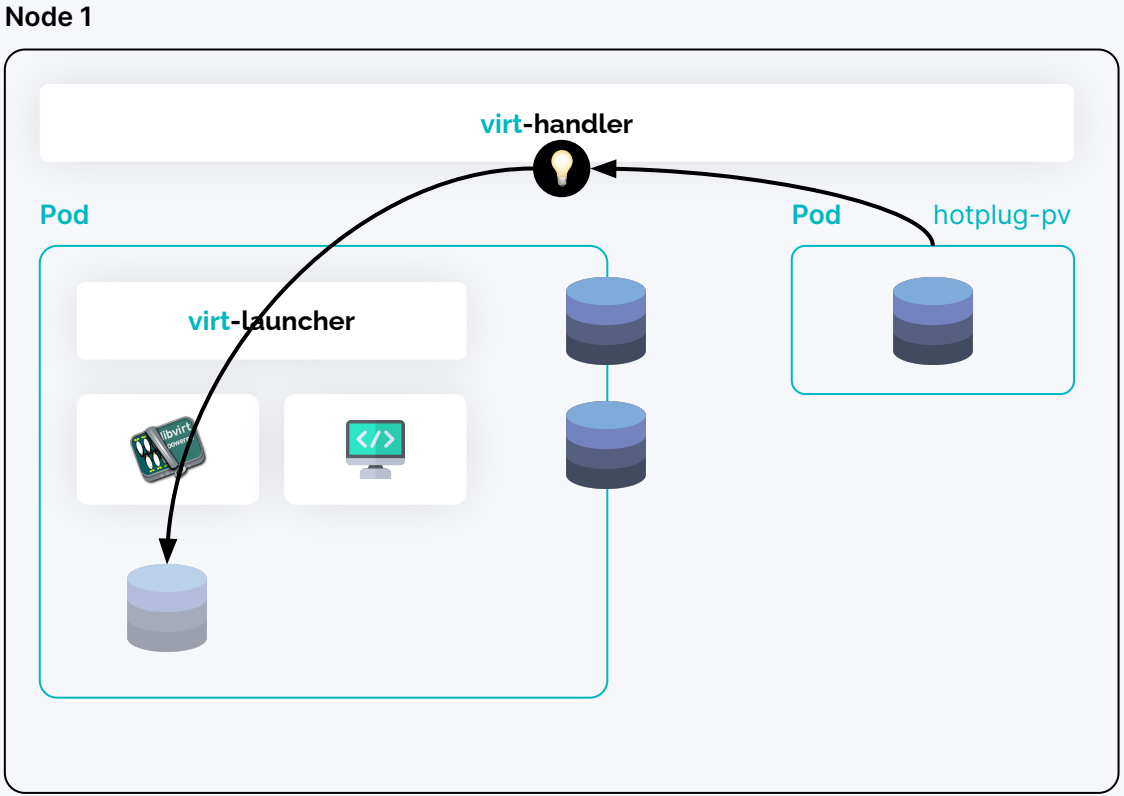
Kubernetes Storage: hotplugging volumes



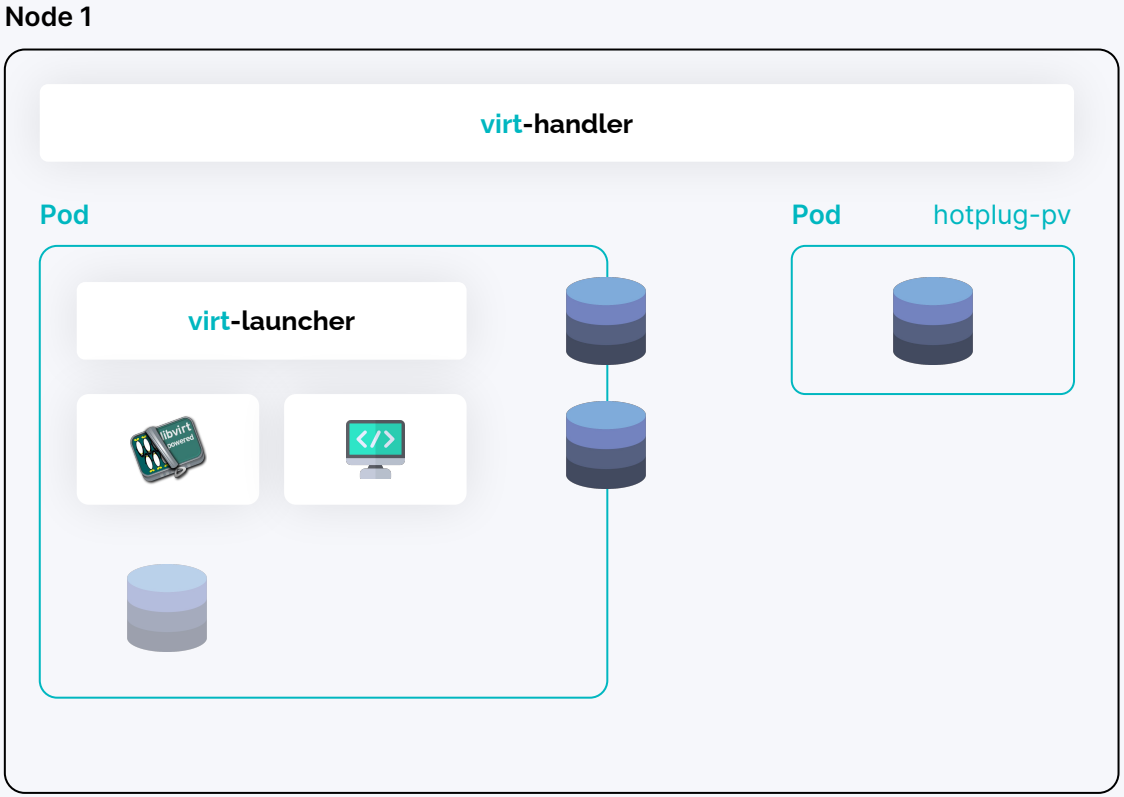
Kubernetes Storage: hotplugging volumes



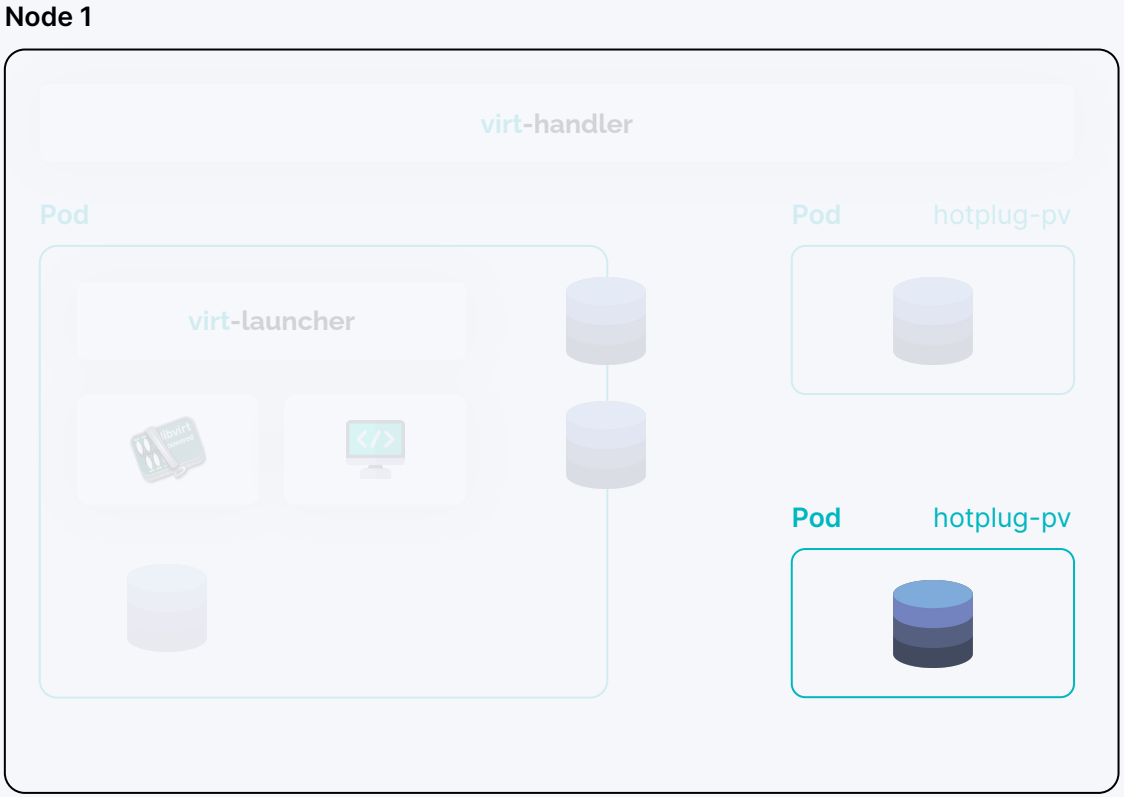
Kubernetes Storage: hotplugging volumes



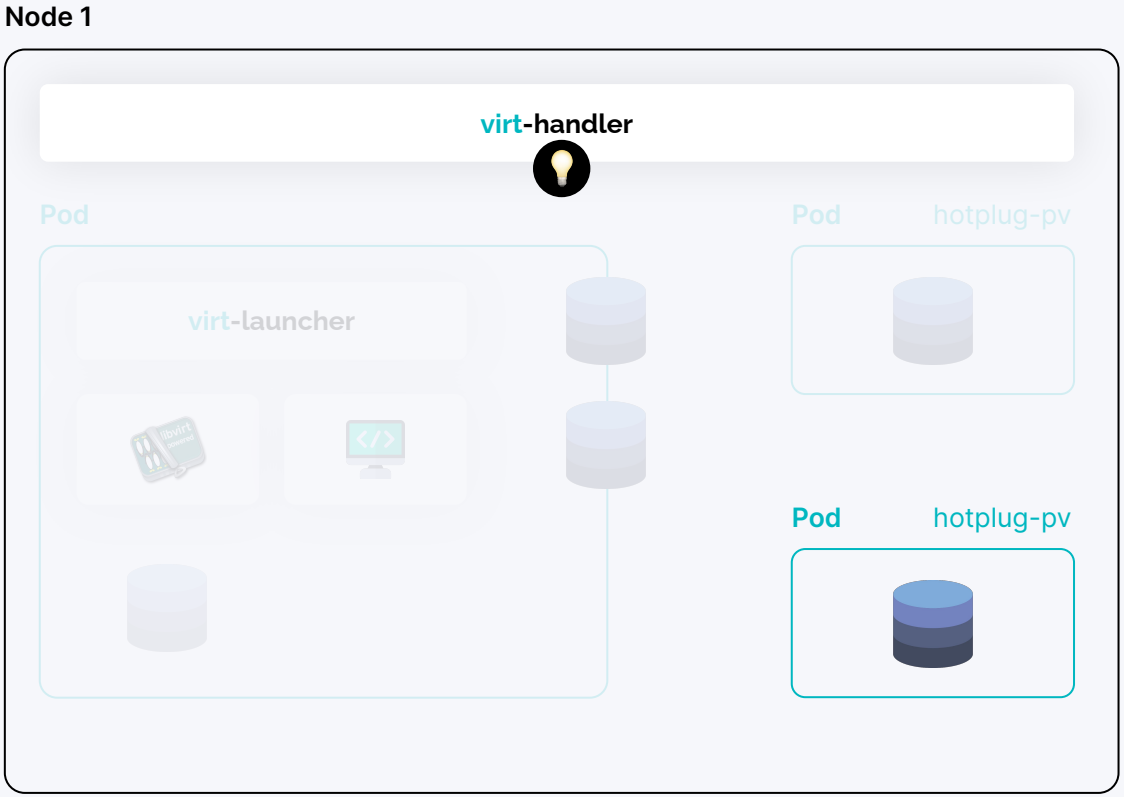
Kubernetes Storage: hotplugging volumes



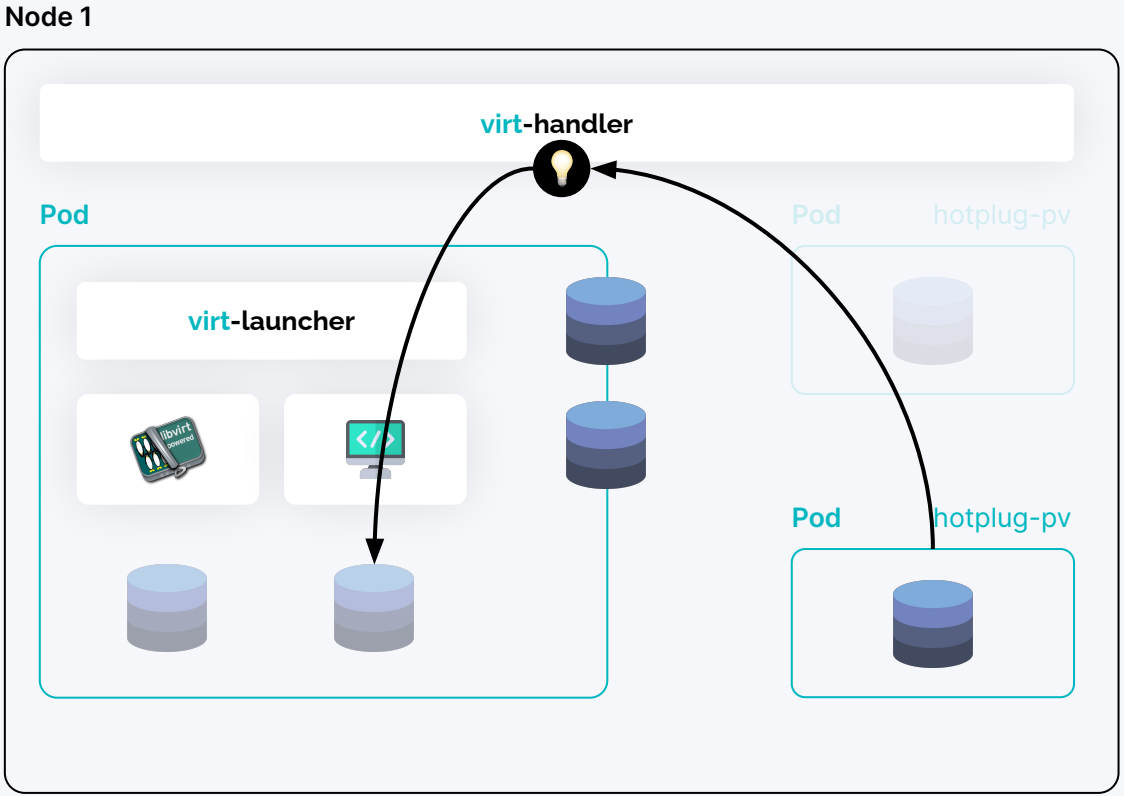
Kubernetes Storage: hotplugging volumes



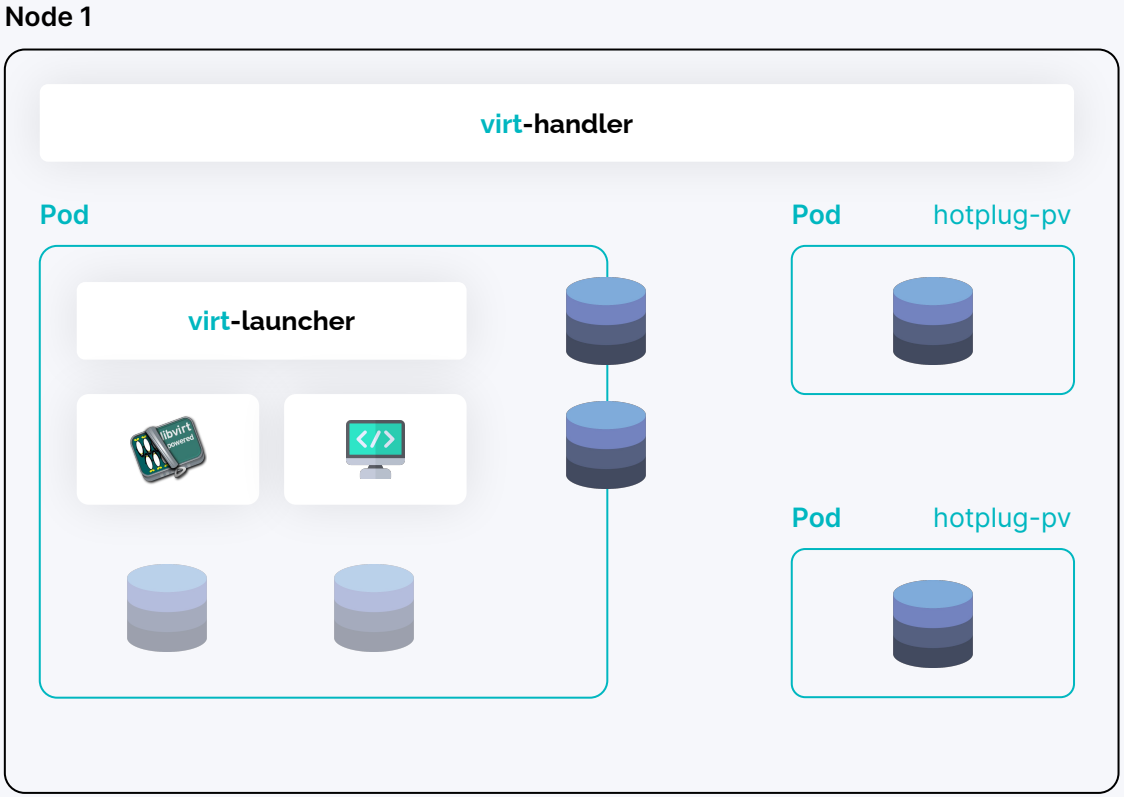
Kubernetes Storage: hotplugging volumes



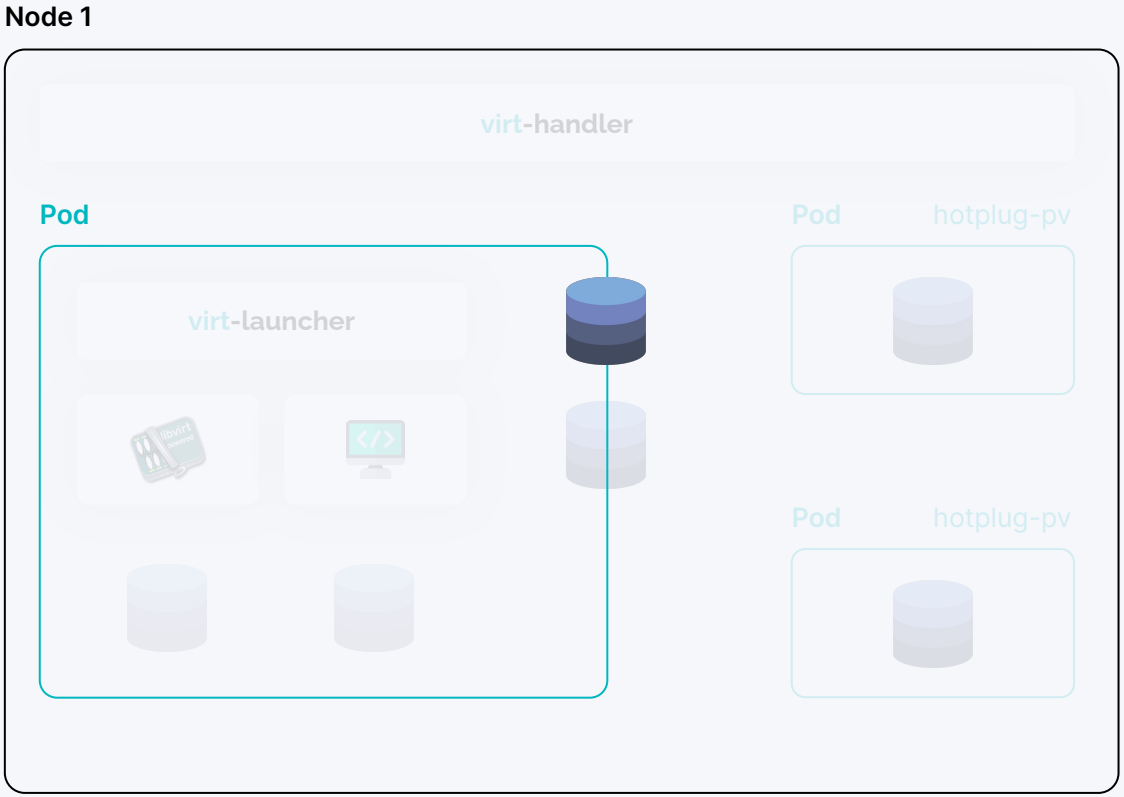
Kubernetes Storage: hotplugging volumes



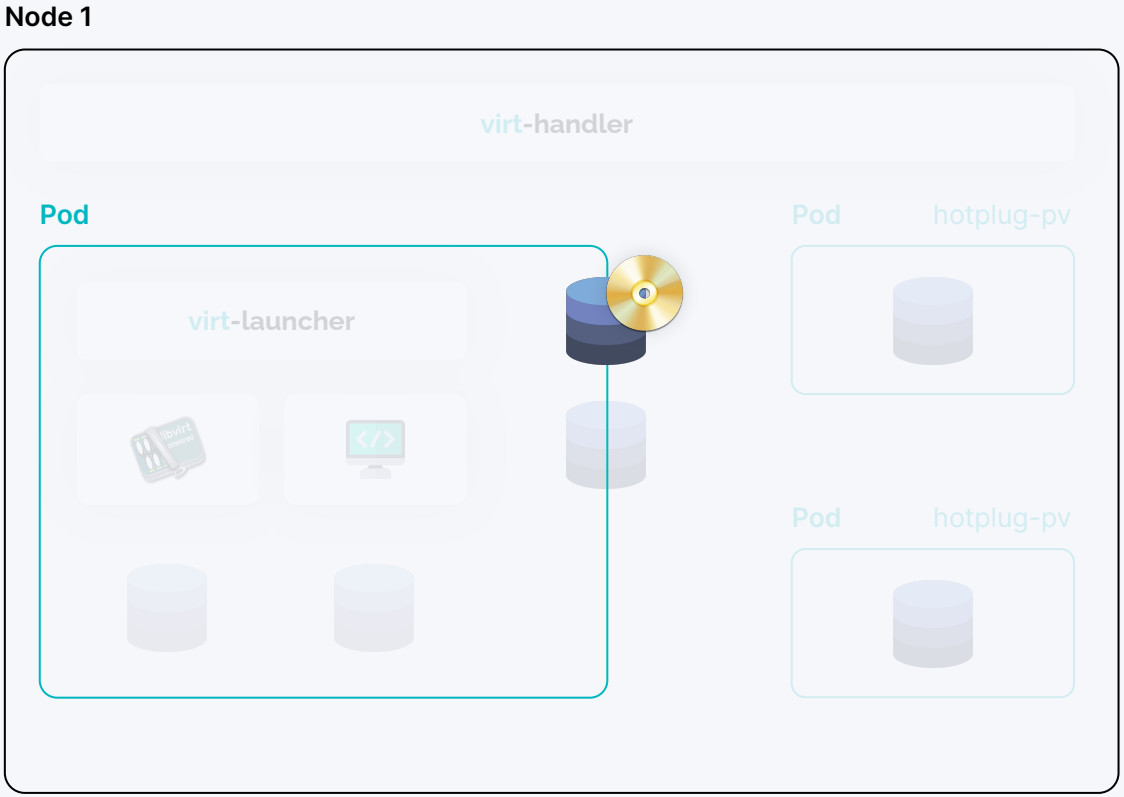
Kubernetes Storage: hotplugging volumes



Kubernetes Storage: hotplugging volumes



Kubernetes Storage: hotplugging volumes



Containerized Data Importer (CDI)

Containerized Data Importer (CDI)

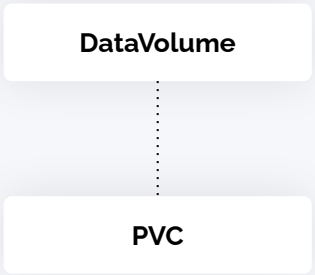
DataVolume

Containerized Data Importer (CDI)

DataVolume

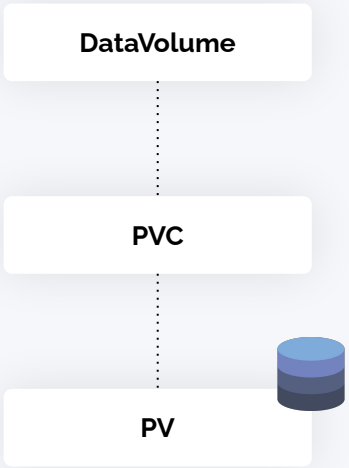
```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)



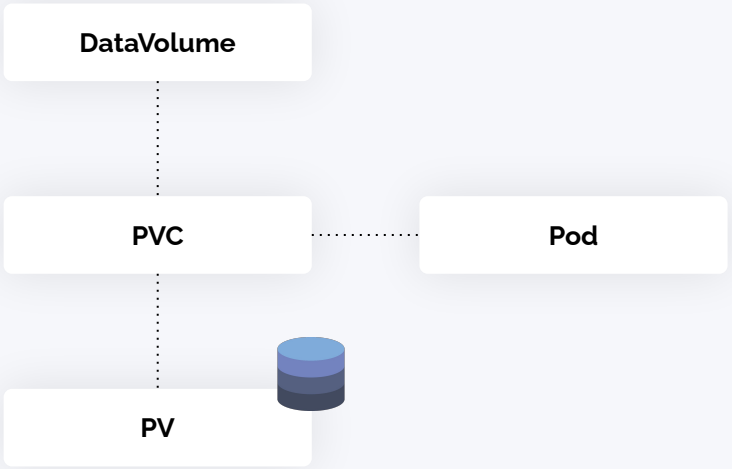

```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)



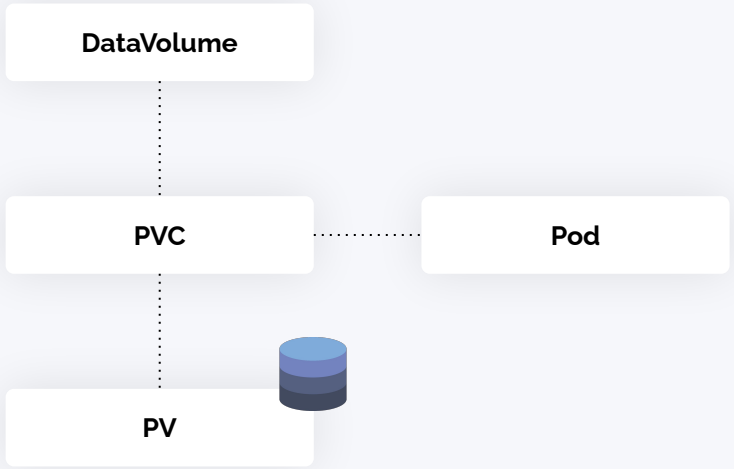

```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)



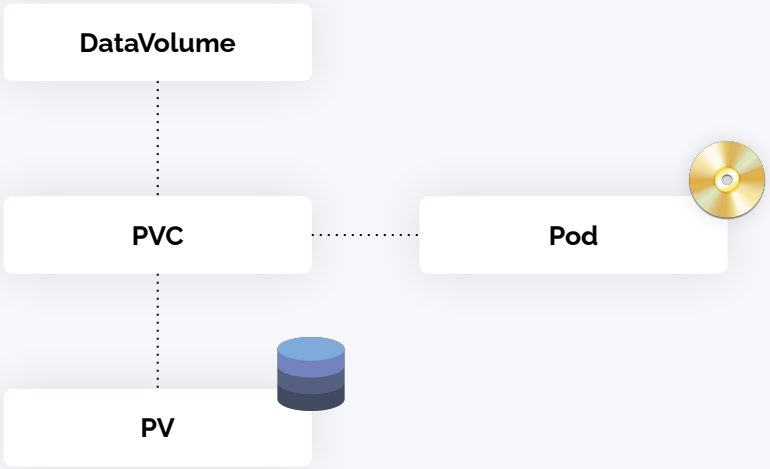

```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)



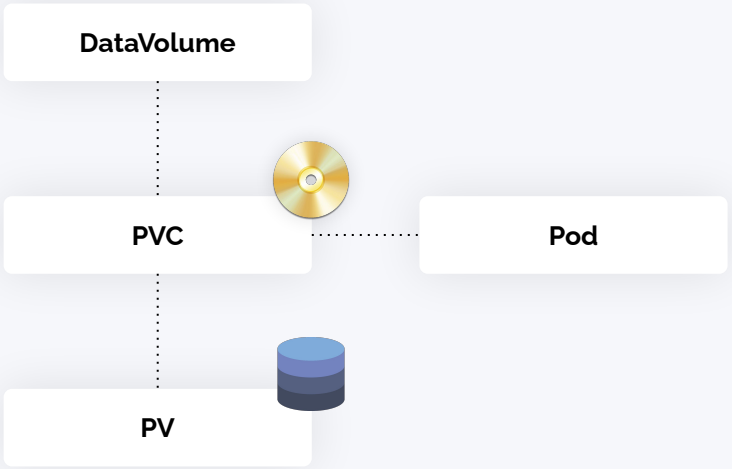

```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
       url: https://server/cloud.img
```

Containerized Data Importer (CDI)



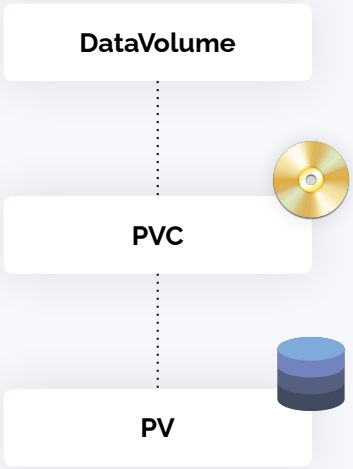

```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
     requests:
        storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
    source:
      http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)



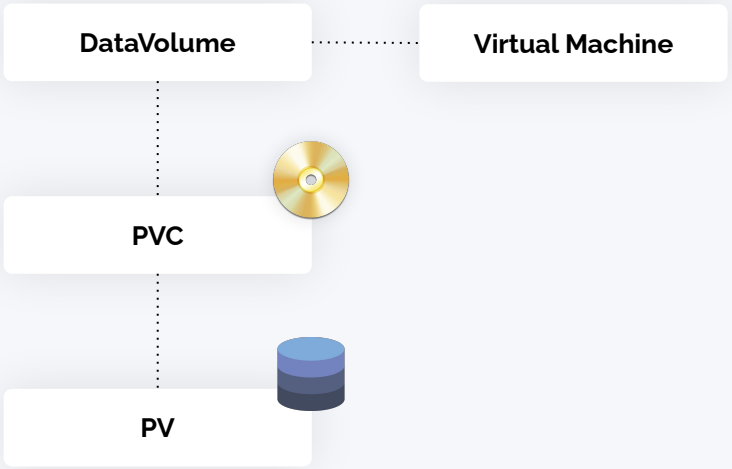

```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
     requests:
        storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
    source:
      http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)




```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
      url: https://server/cloud.img
```

Containerized Data Importer (CDI)




```
pvc:
  accessModes:
  - ReadWriteMany
  resources:
      requests:
          storage: 10Gi
  storageClassName: linstor-thindata-r2
  volumeMode: Block
     source:
       http:
      url: https://server/cloud.img
```

How does this work?

VM

1

Kubernetes Runtime

CRI

2

Kubernetes Storage

CSI

3

Kubernetes networking

CNI

How does this work?

VM

1

Kubernetes Runtime

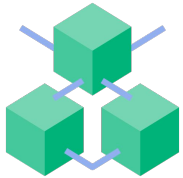
CRI

2

Kubernetes Storage

CSI

3

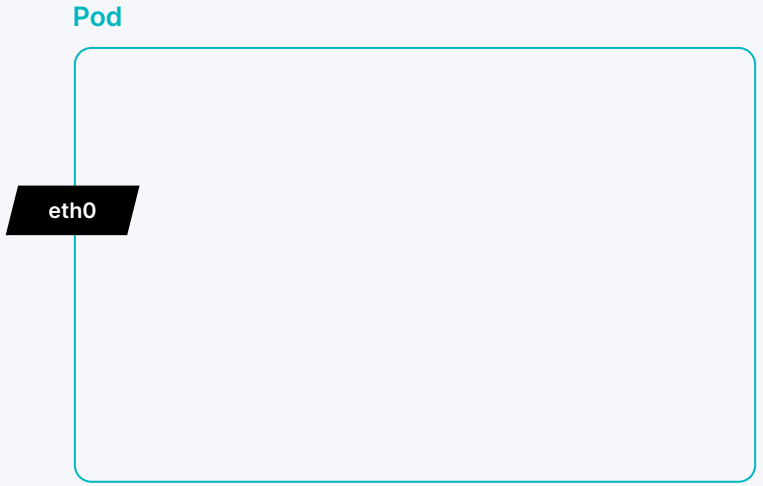


Kubernetes networking

CNI

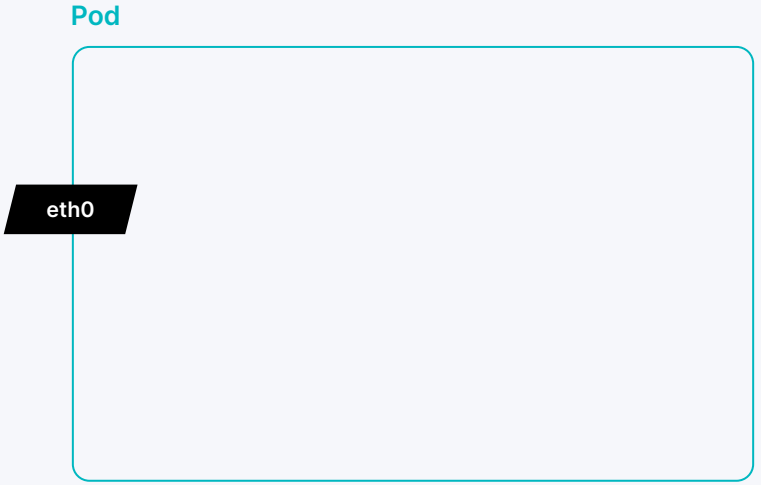
Kubernetes networking

Kubernetes networking



Kubernetes networking

Backend

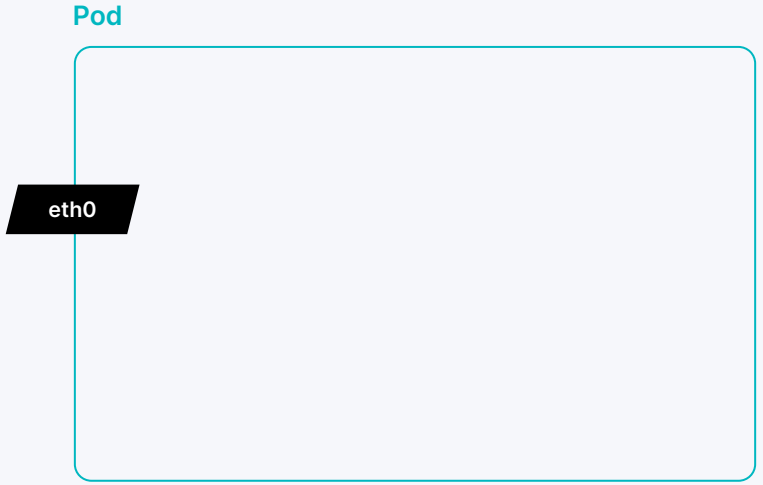


Kubernetes networking

Backend

pod

Default Kubernetes network

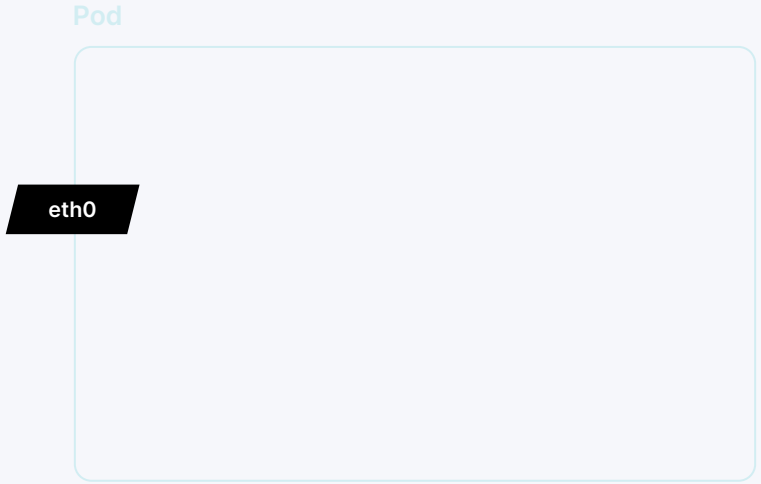


Kubernetes networking

Backend

pod

Default Kubernetes network



Kubernetes networking

Backend

pod

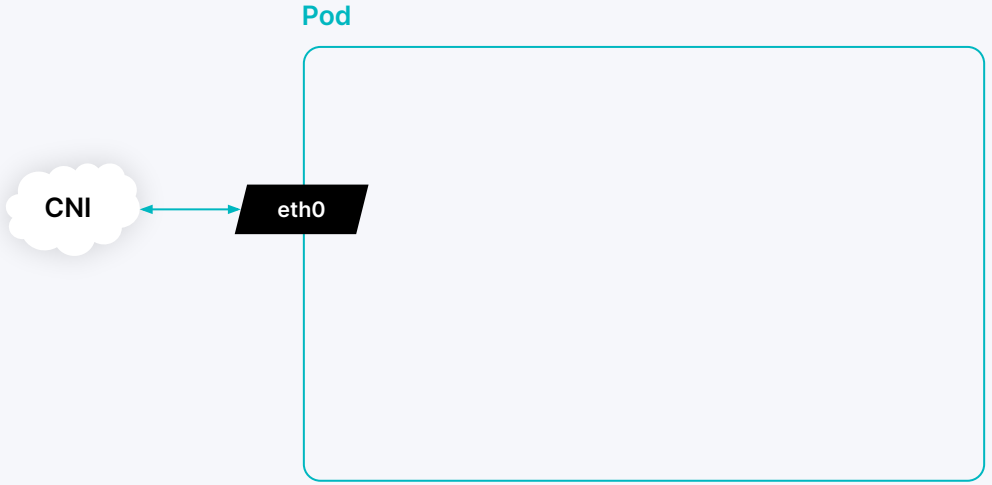
Default Kubernetes network

Kubernetes networking

Backend

pod

Default Kubernetes network



Kubernetes networking

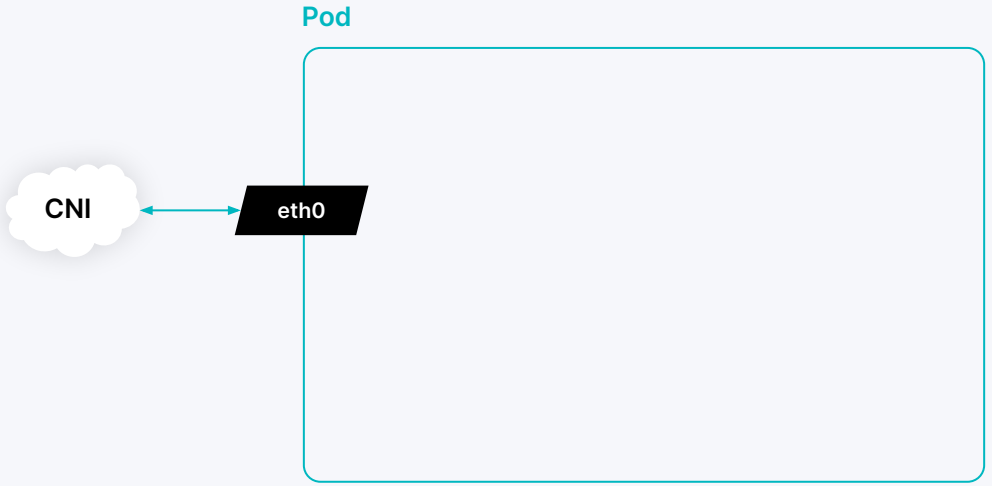
Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus



Kubernetes networking

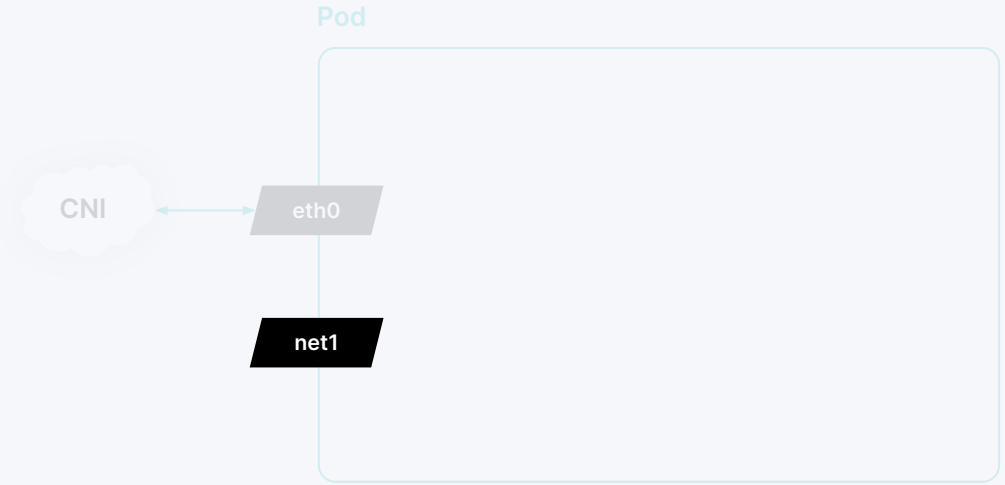
Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus



Kubernetes networking

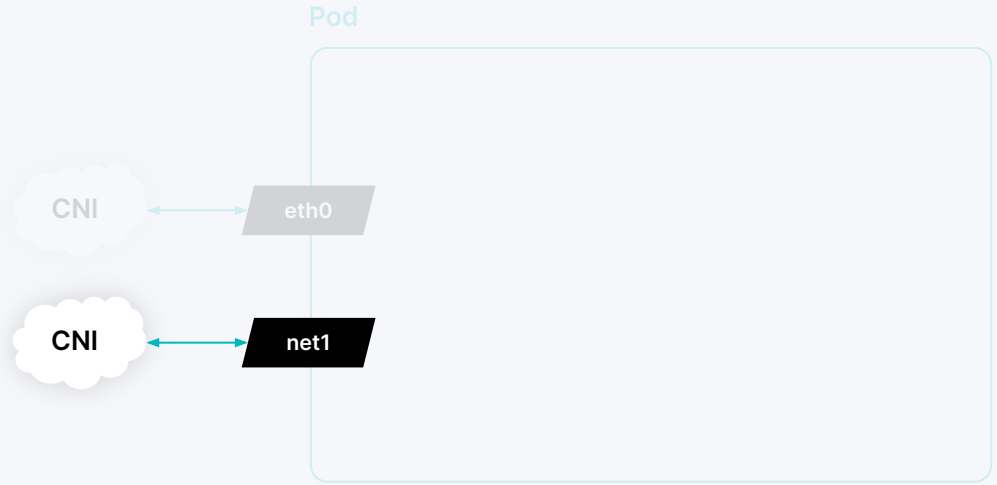
Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus



Kubernetes networking

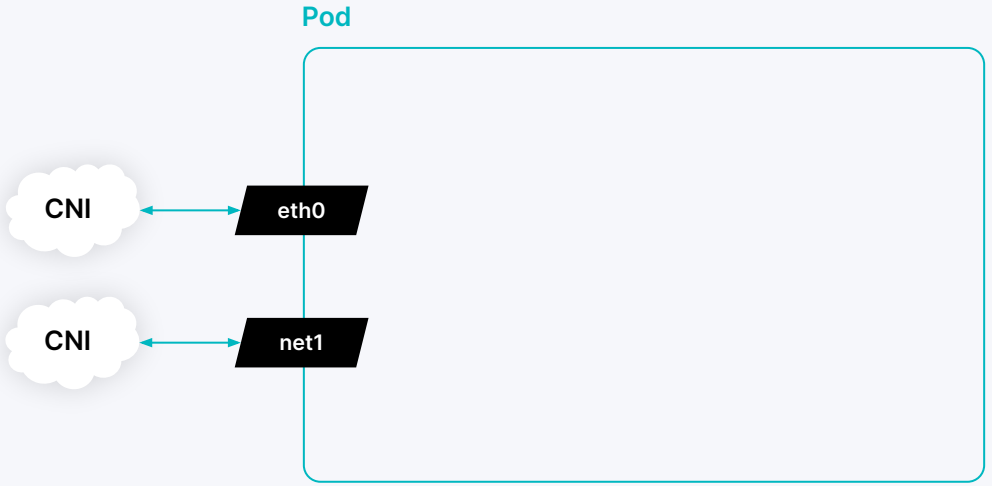
Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus



Kubernetes networking

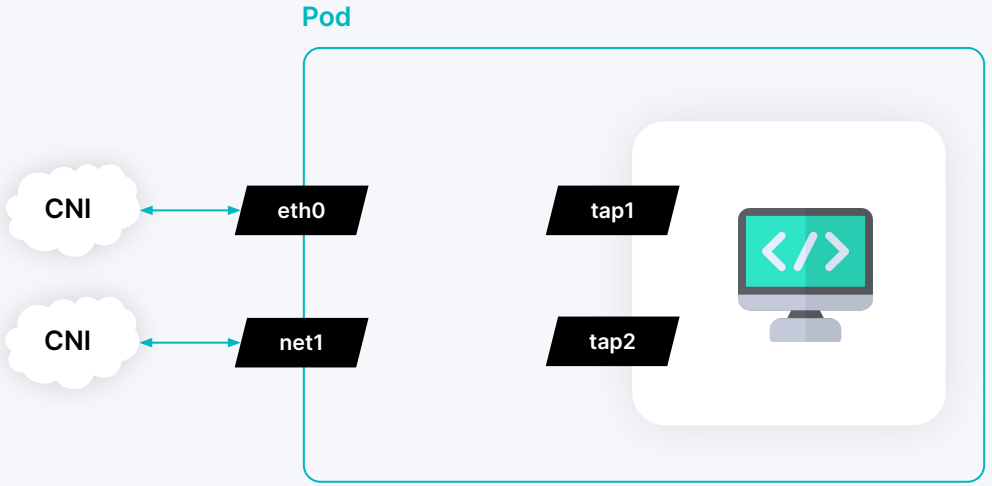
Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus



Kubernetes networking

Backend

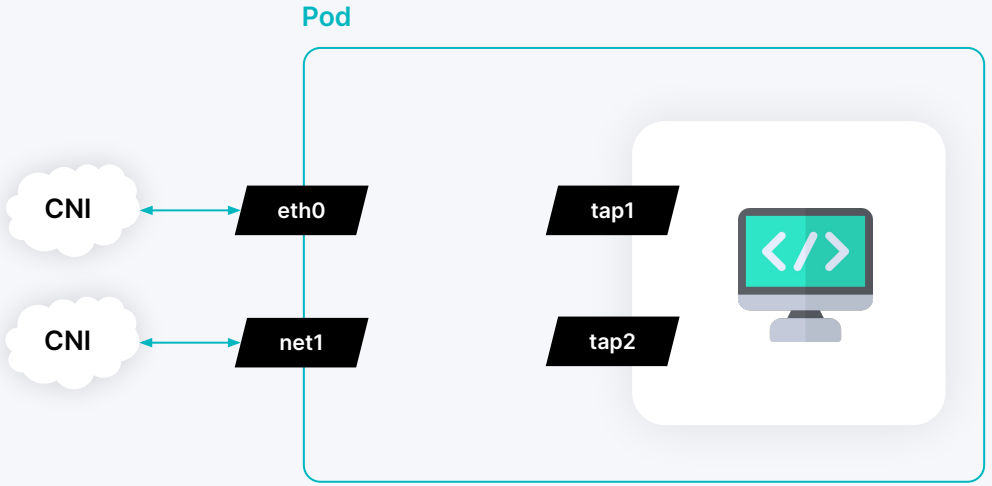
pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend



Kubernetes networking

Backend

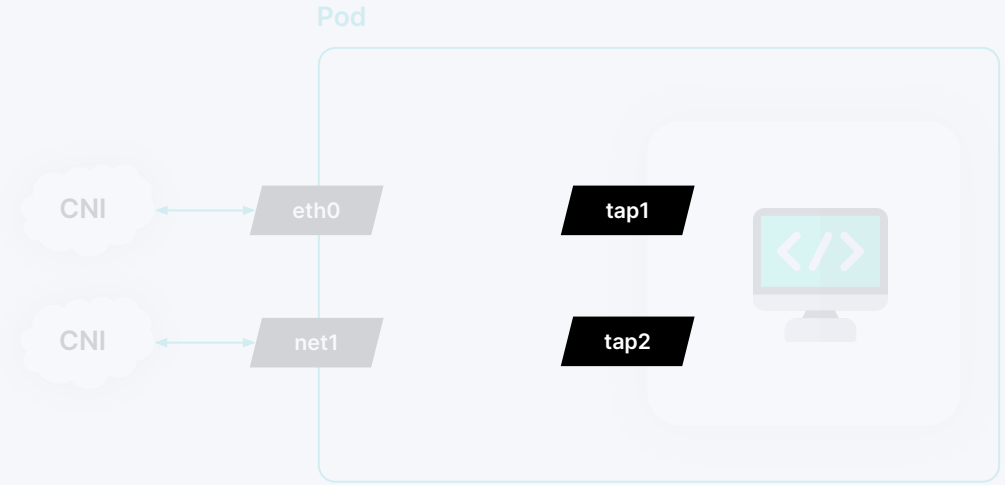
pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend



Kubernetes networking

Backend

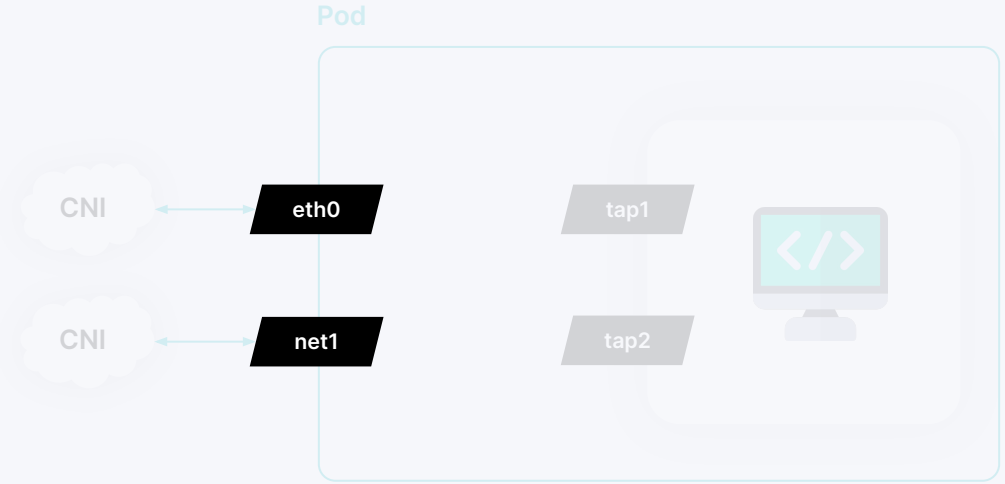
pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend



Kubernetes networking

Backend

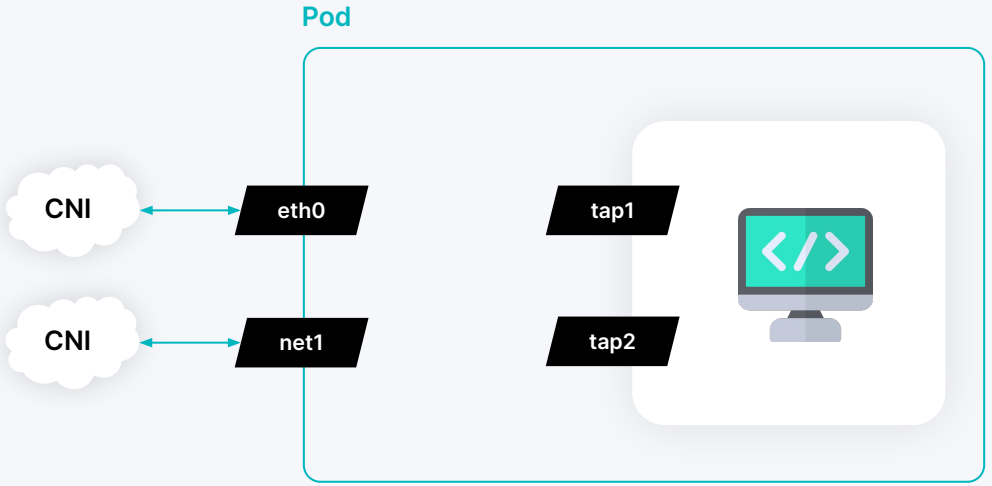
pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend



Kubernetes networking

Backend

pod

Default Kubernetes network

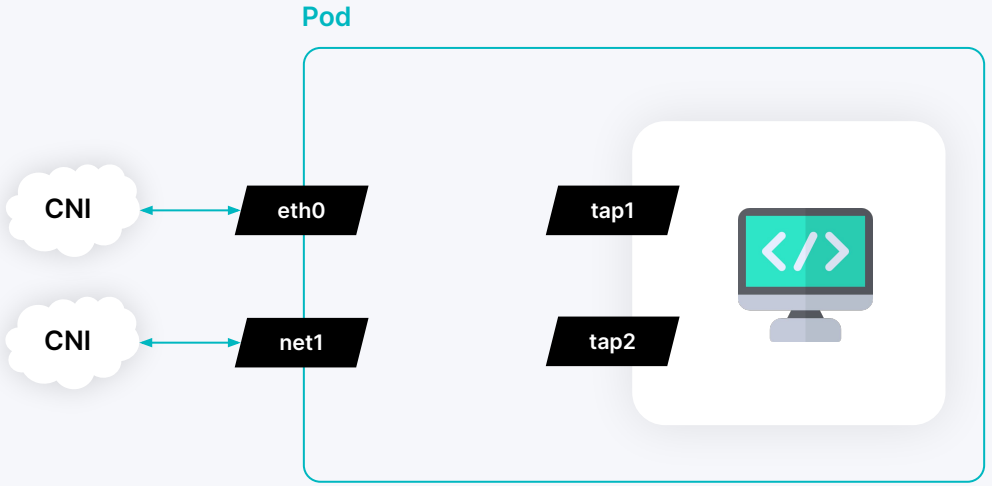
multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic



Kubernetes networking

Backend

pod

Default Kubernetes network

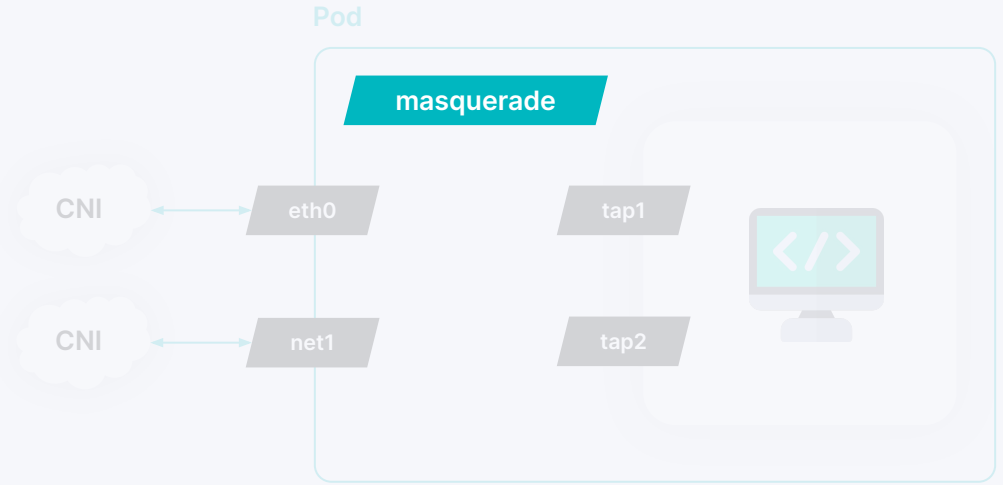
multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic



Kubernetes networking

Backend

pod

Default Kubernetes network

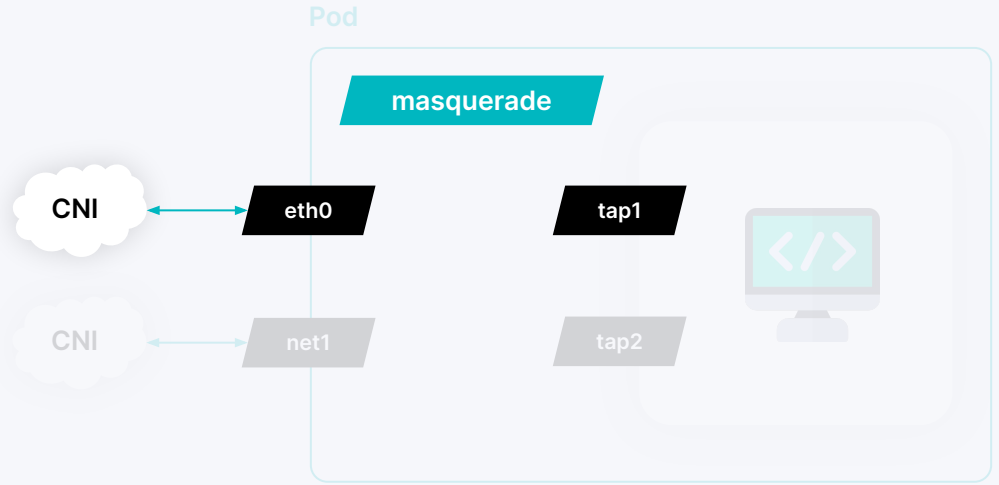
multus

Secondary network provided using Multus

Frontend

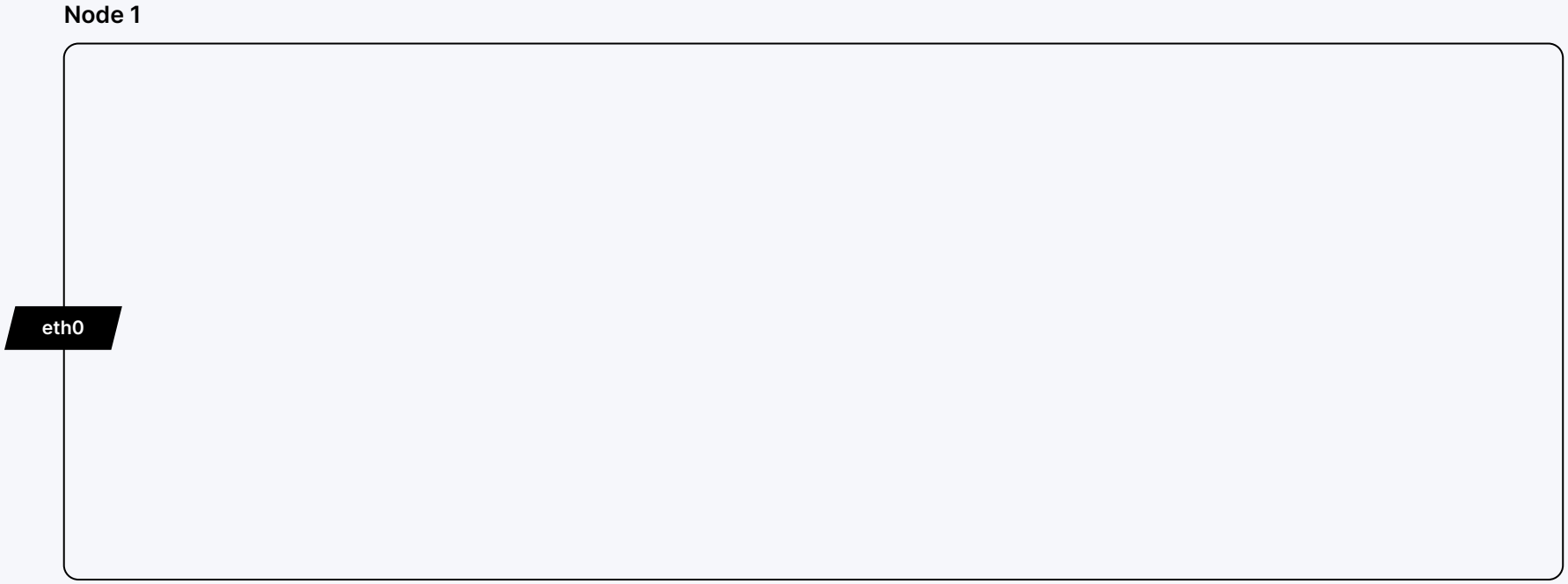
masquerade

Connect using Iptables rules to nat the traffic

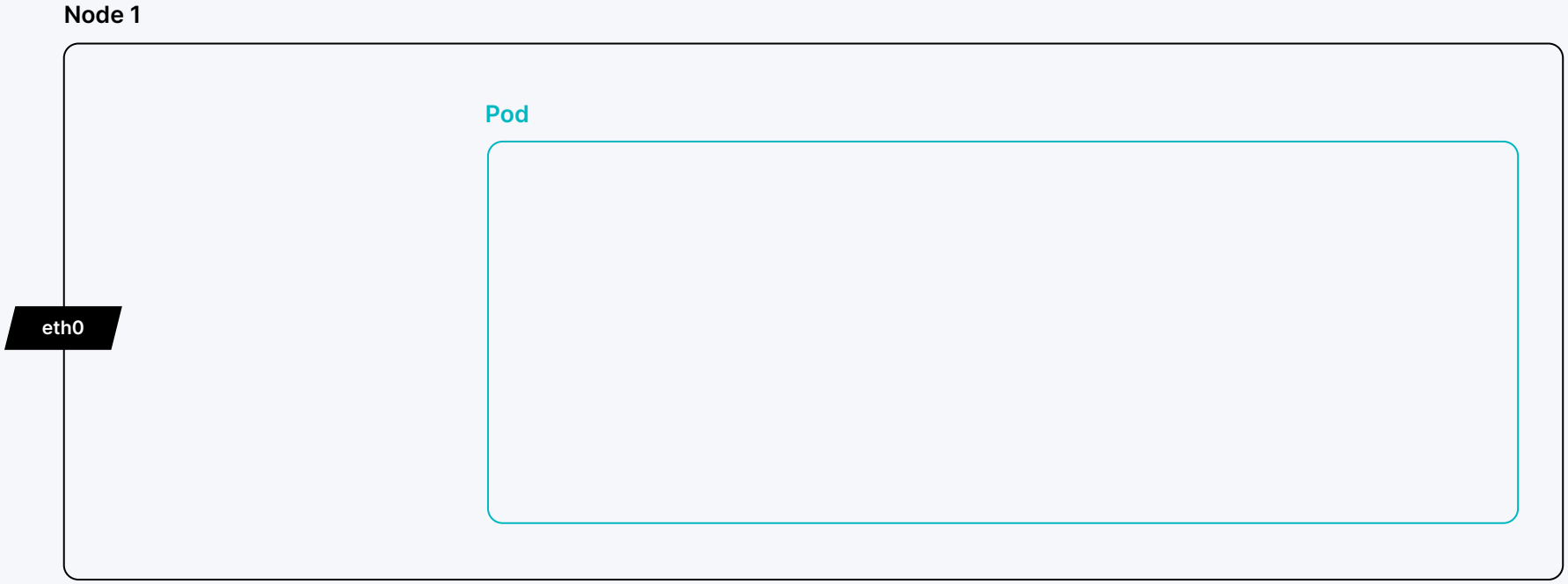


masquerade

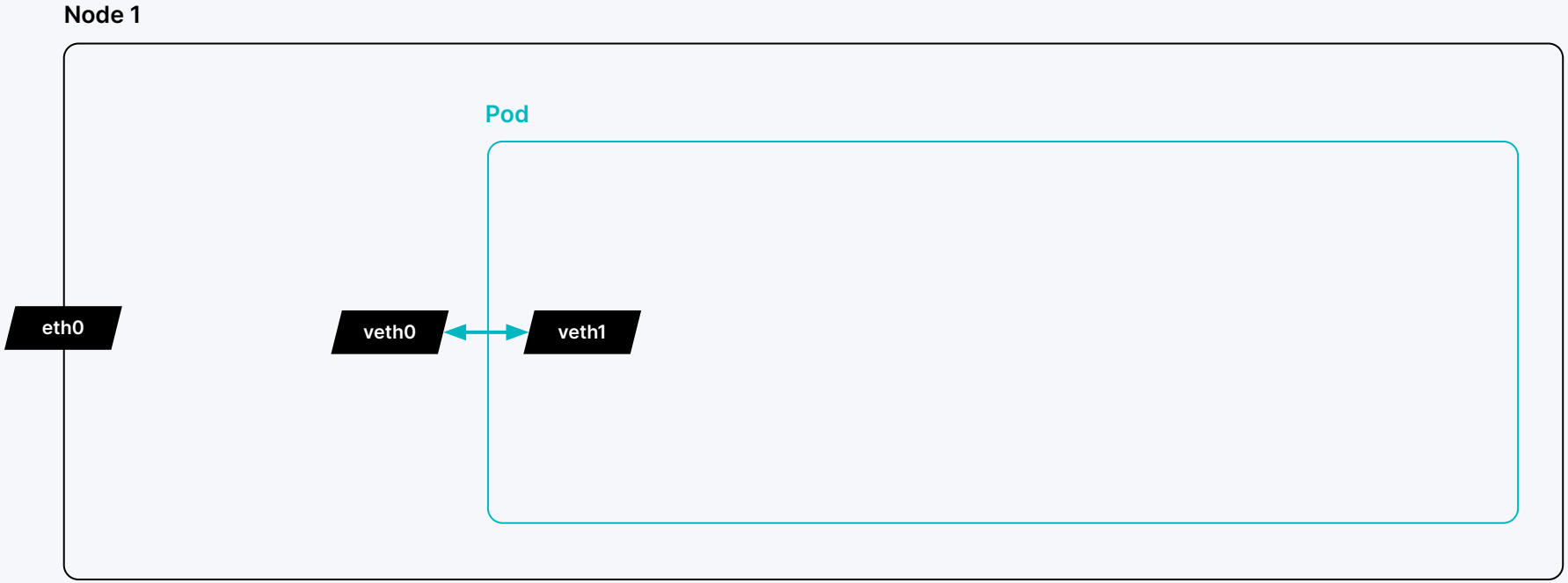
masquerade



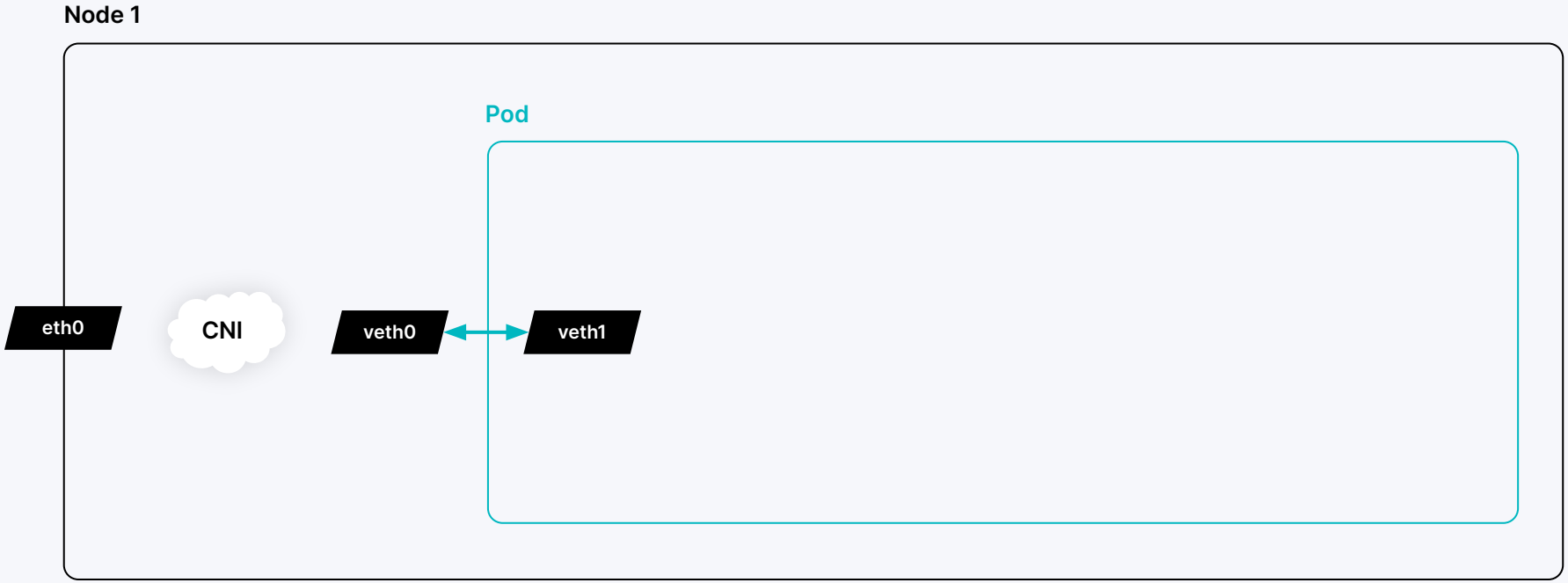
masquerade



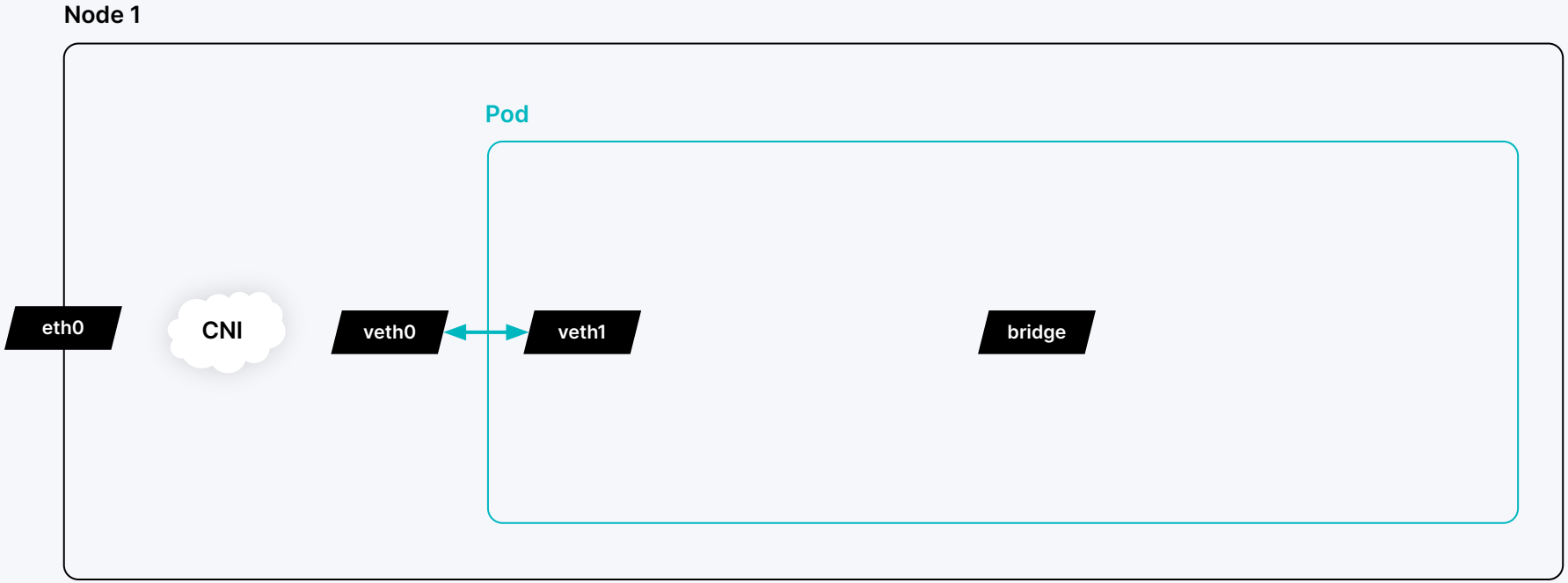
masquerade



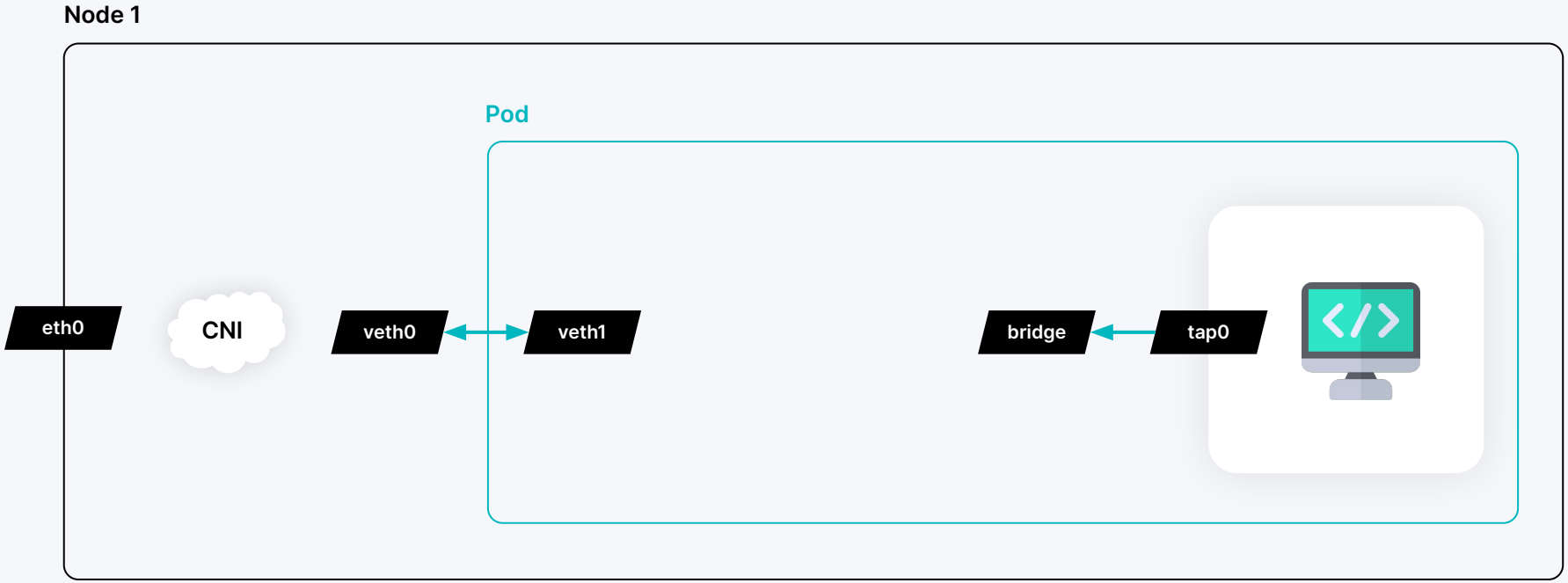
masquerade



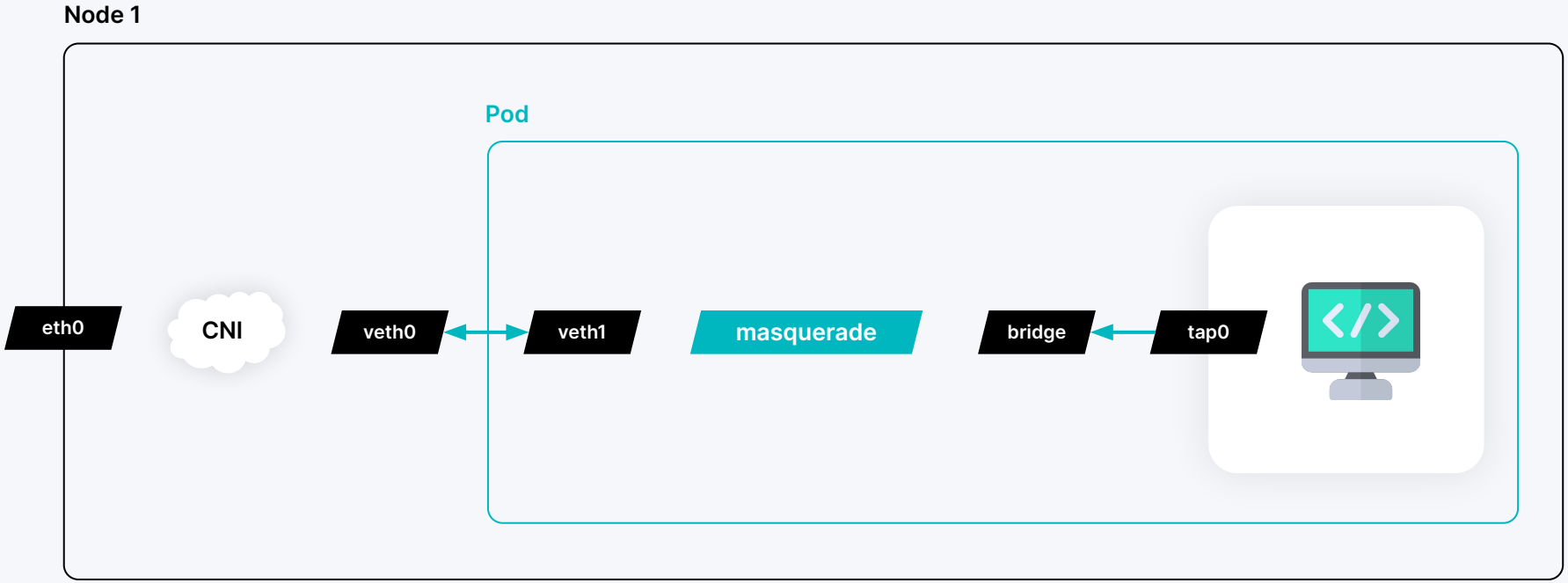
masquerade



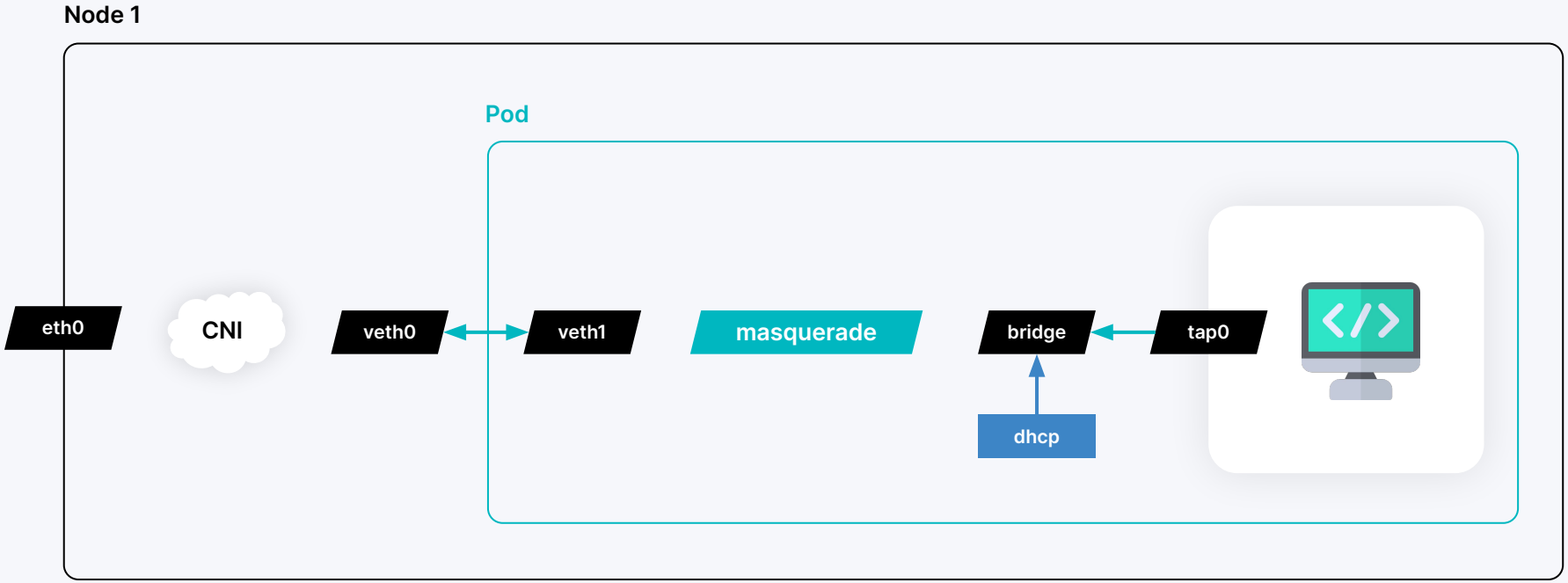
masquerade



masquerade



masquerade



Kubernetes networking

Backend

pod

Default Kubernetes network

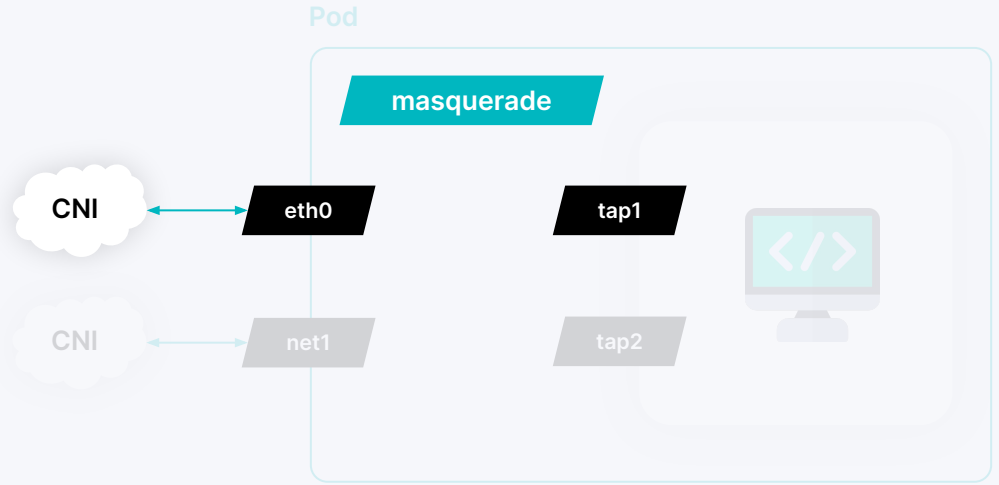
multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic



Kubernetes networking

Backend

pod

Default Kubernetes network

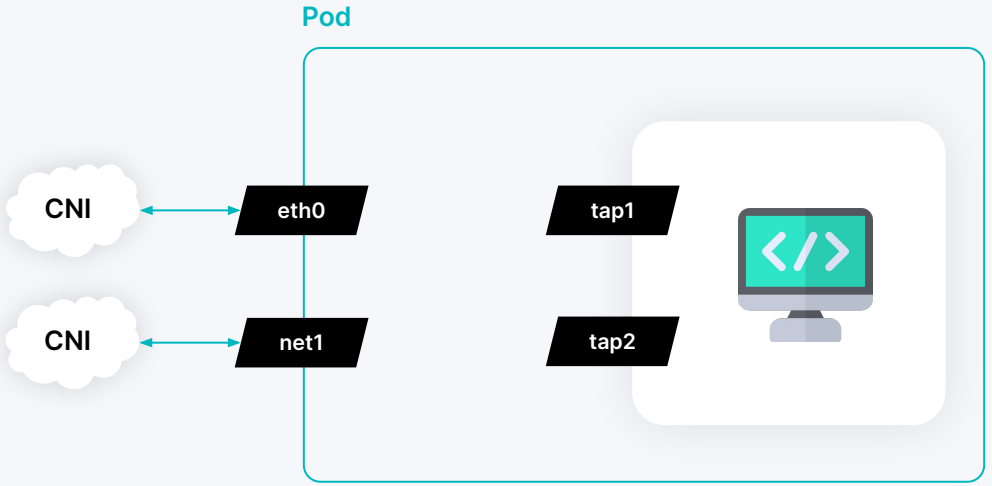
multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

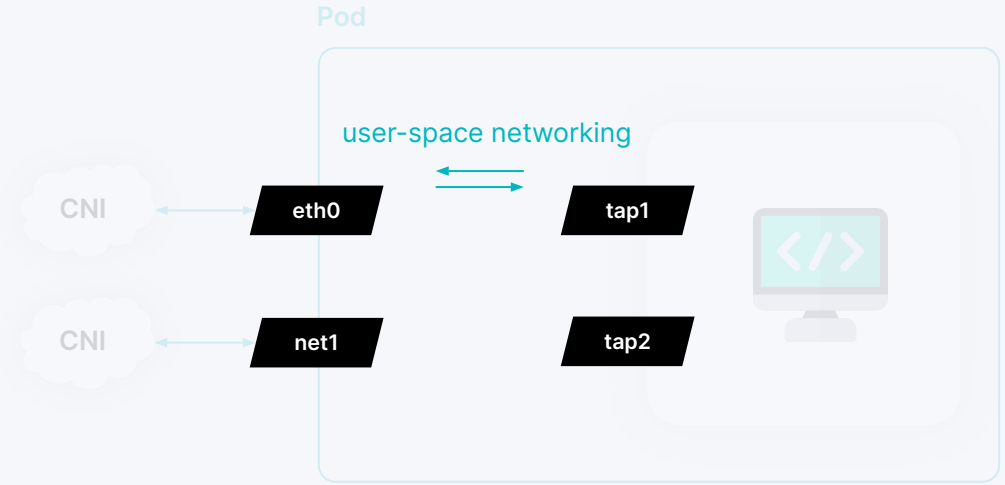
Frontend

masquerade

Connect using Iptables rules to nat the traffic

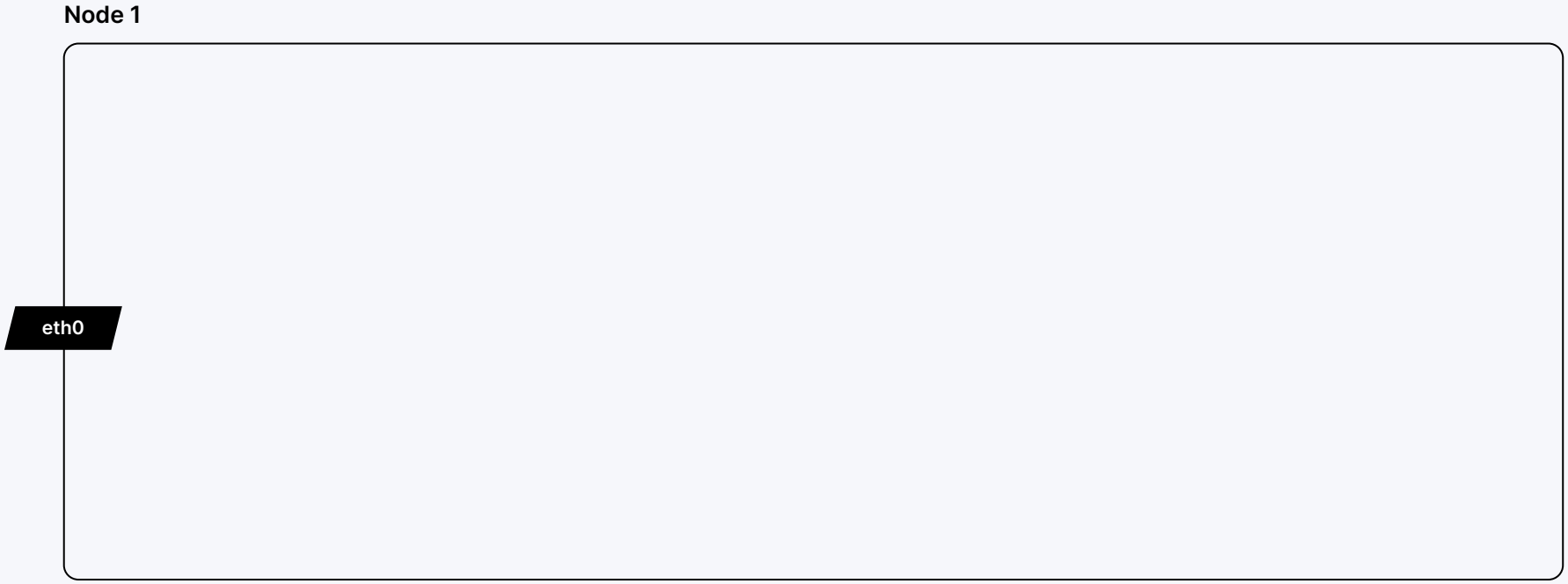
slirp

Connect using QEMU user networking mode

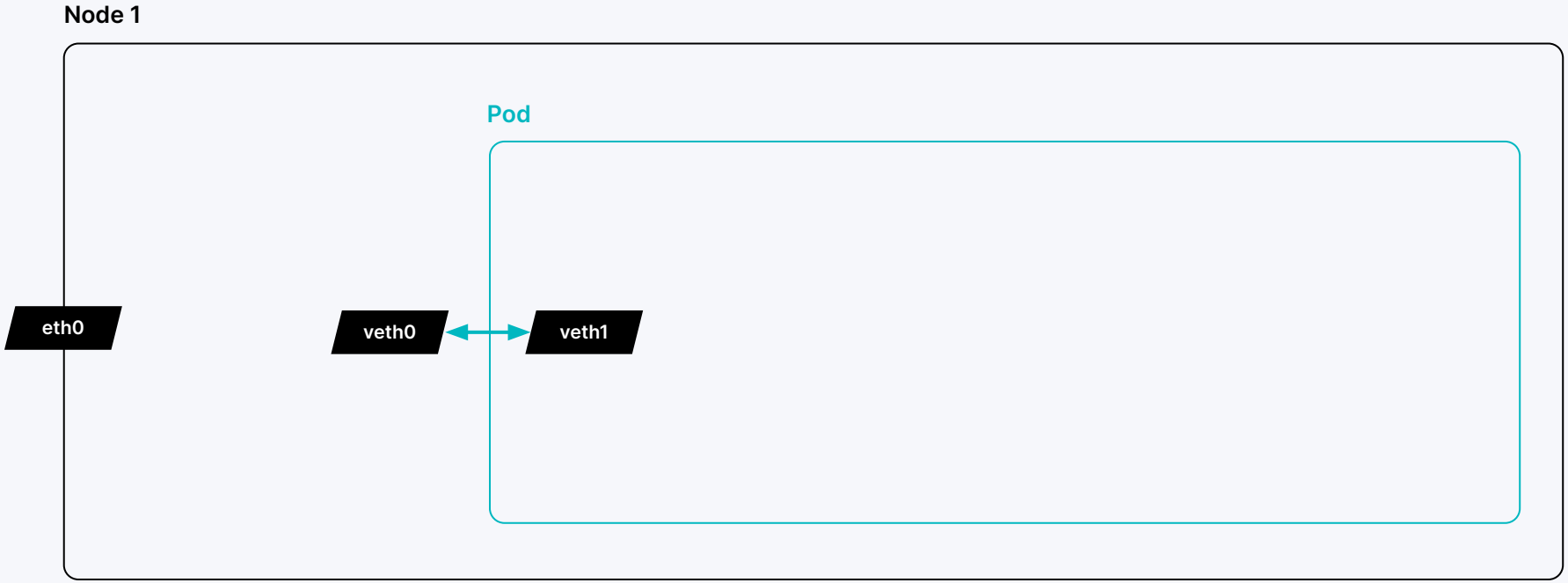


slirp

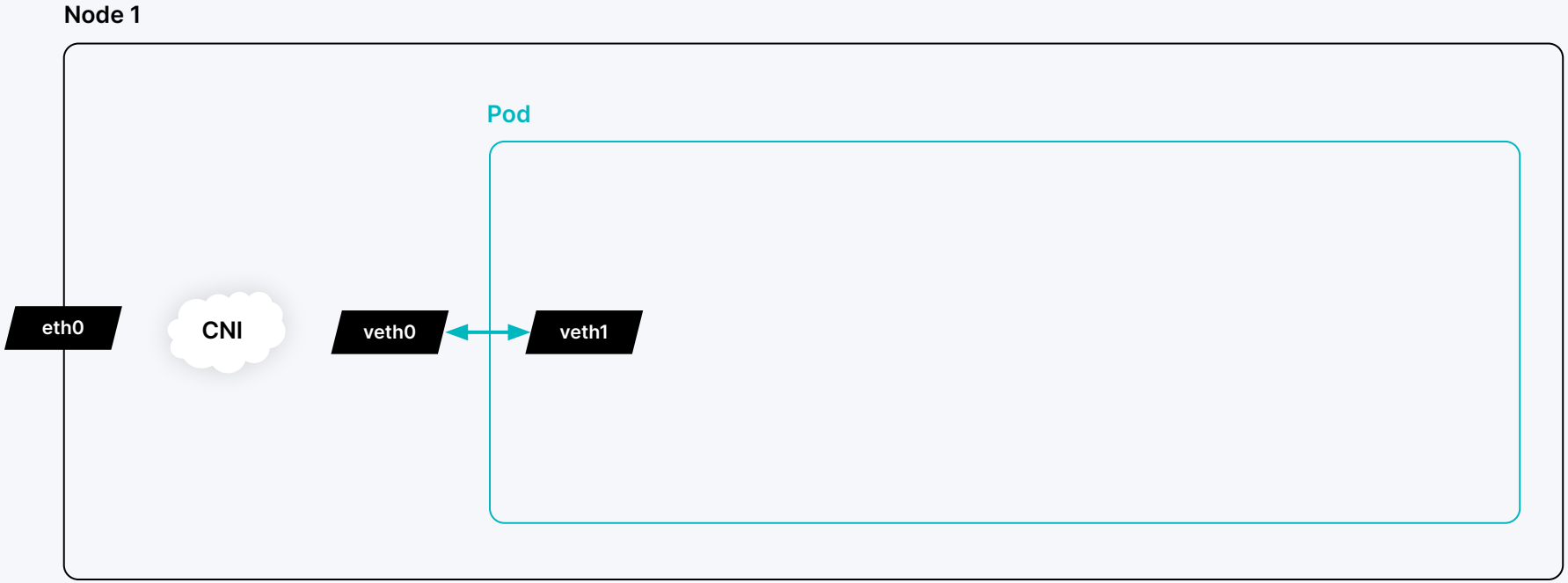
slirp



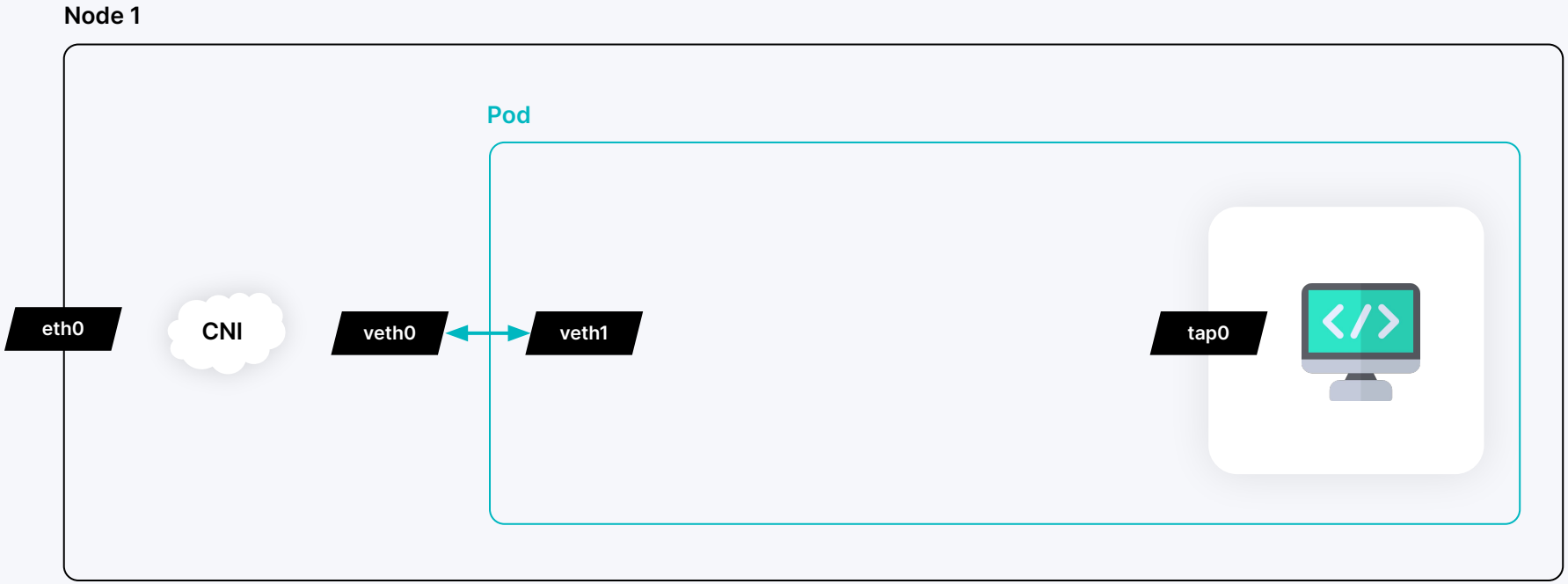
slirp



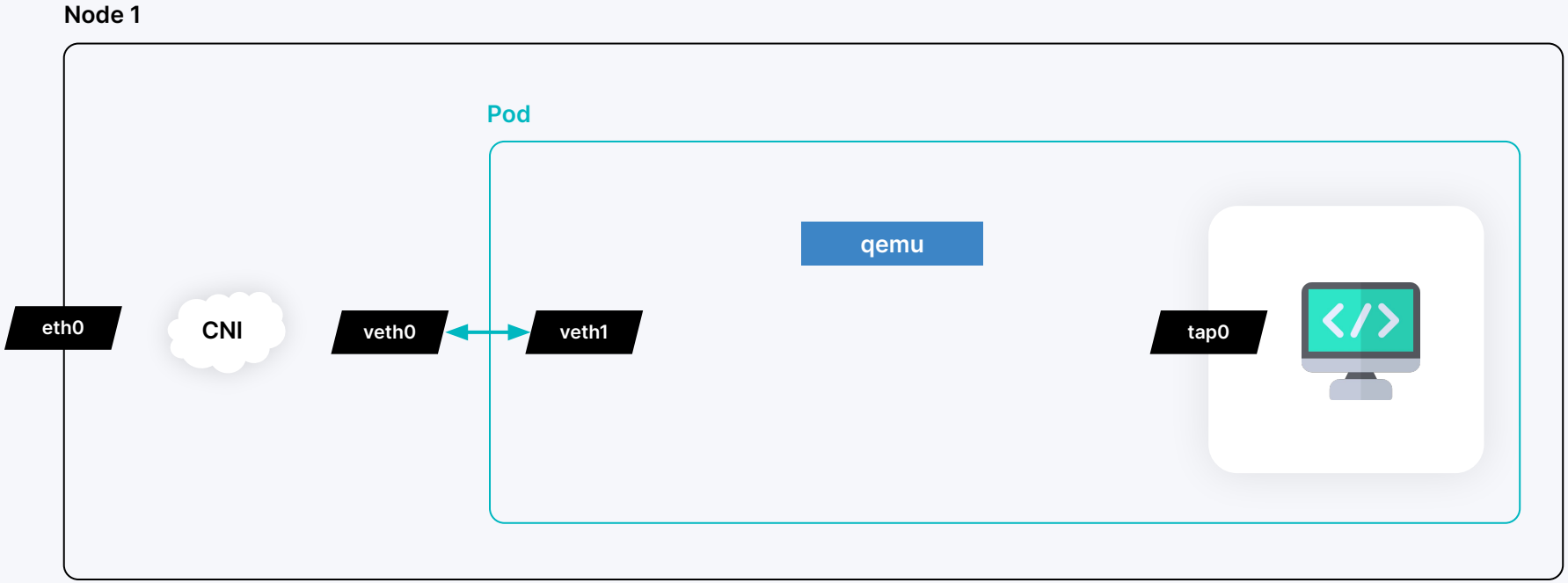
slirp



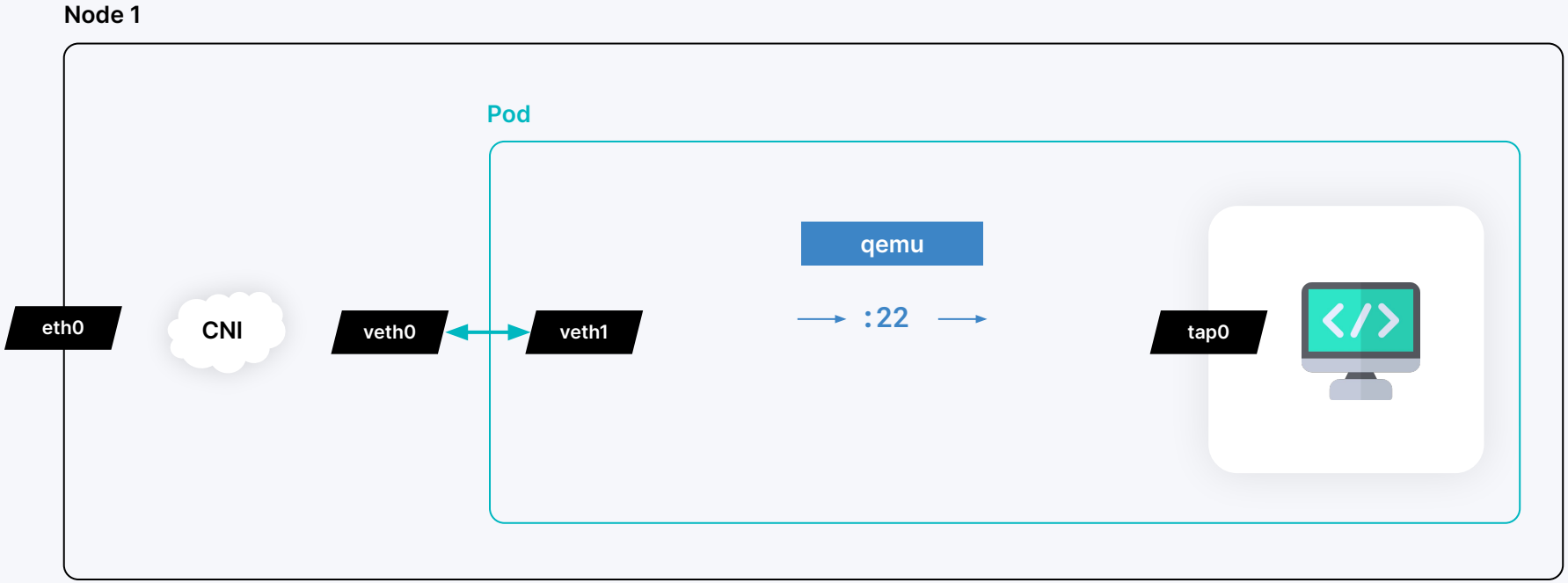
slirp



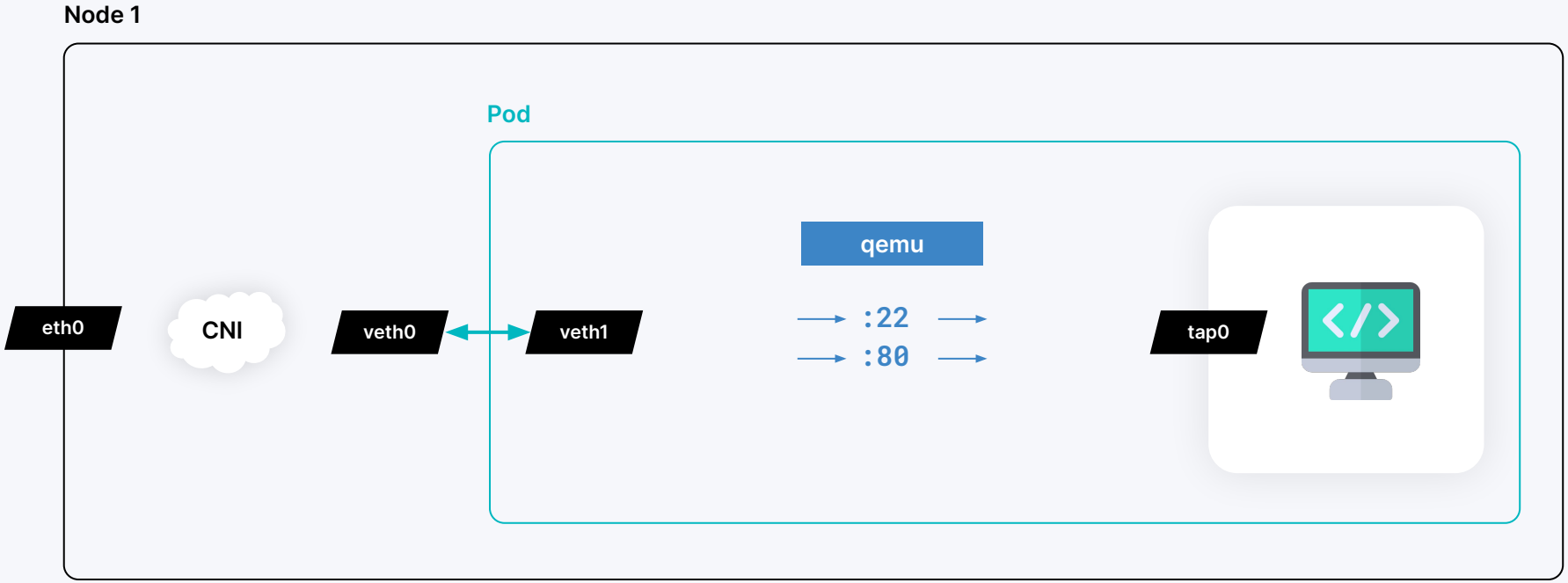
slirp



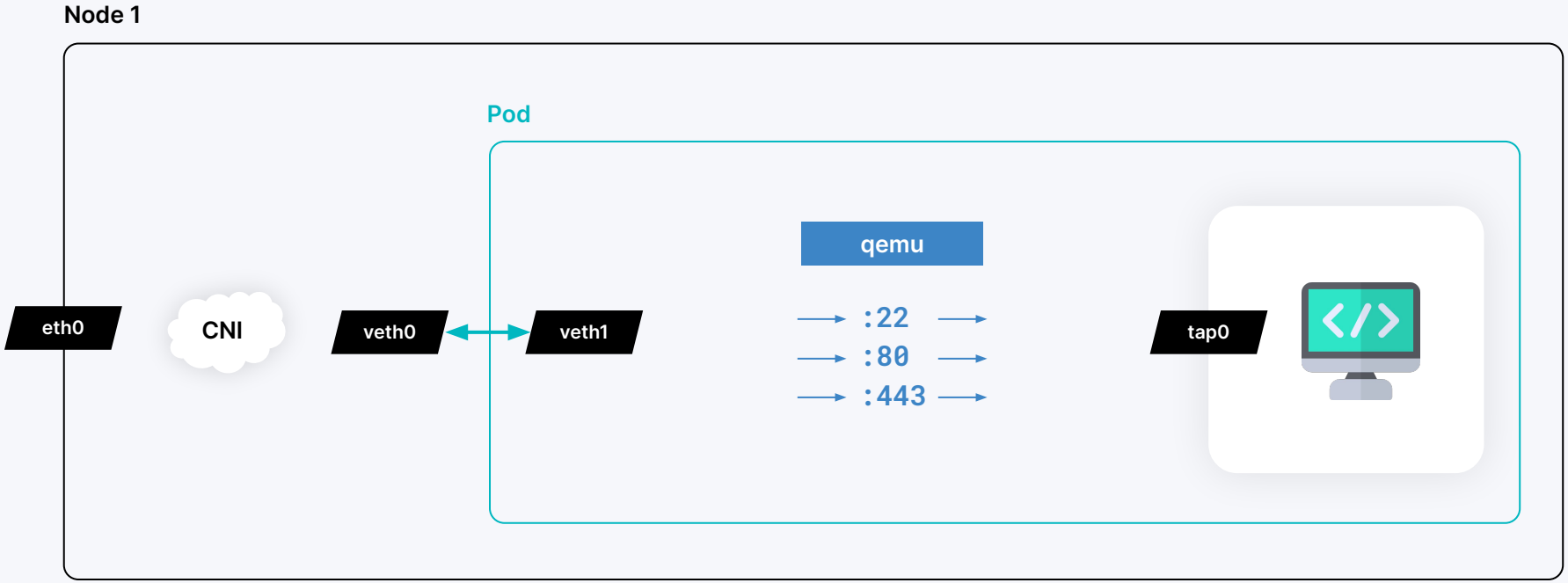
slirp



slirp



slirp



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

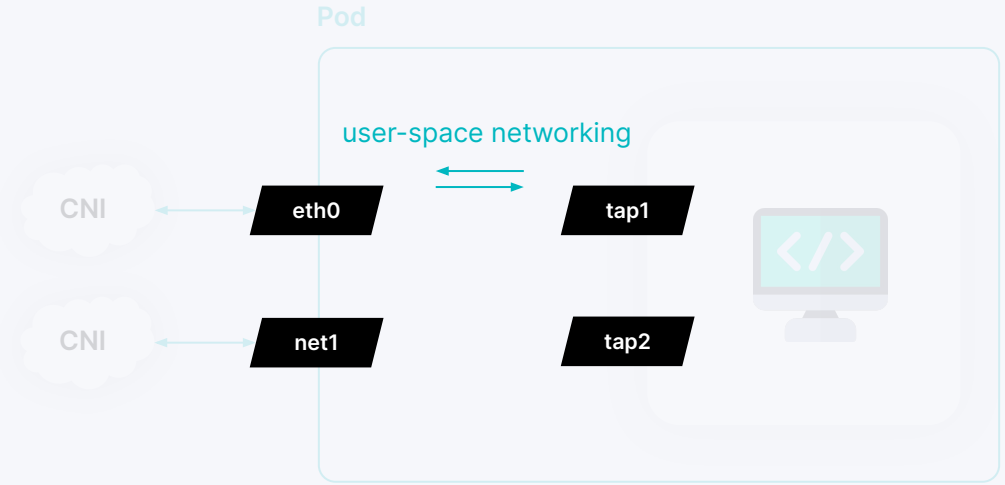
Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

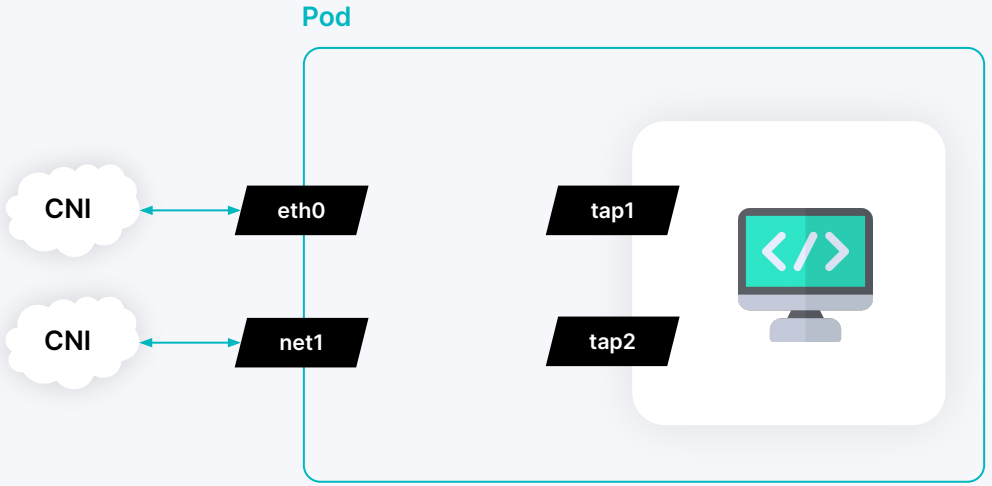
Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

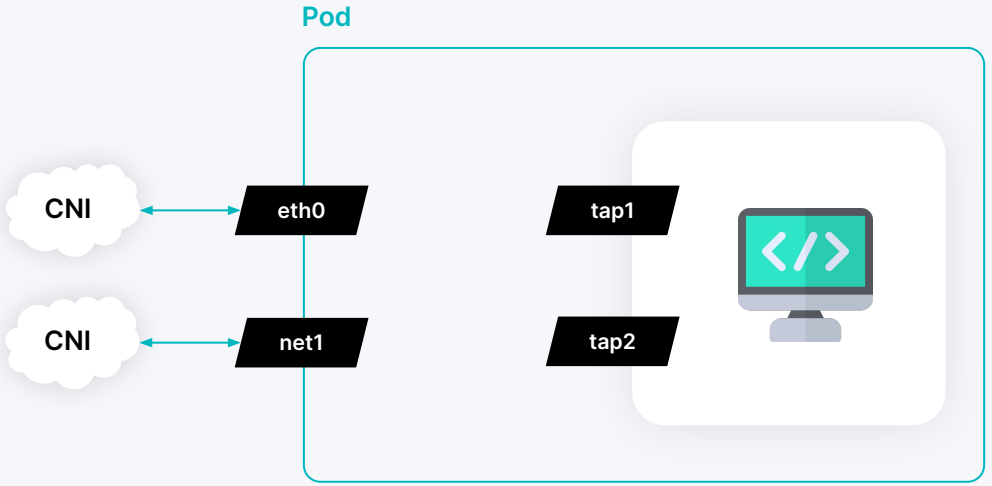
Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

Connect using a linux bridge



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

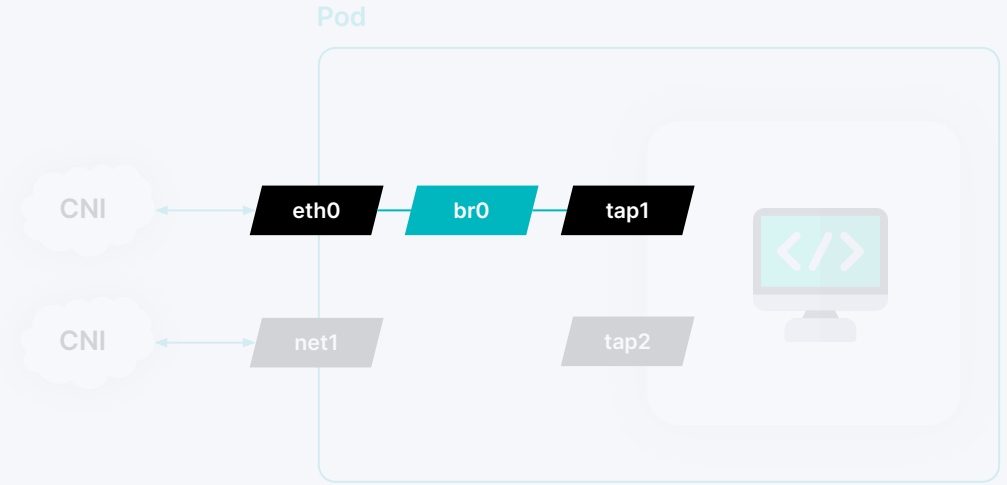
Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

Connect using a linux bridge



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

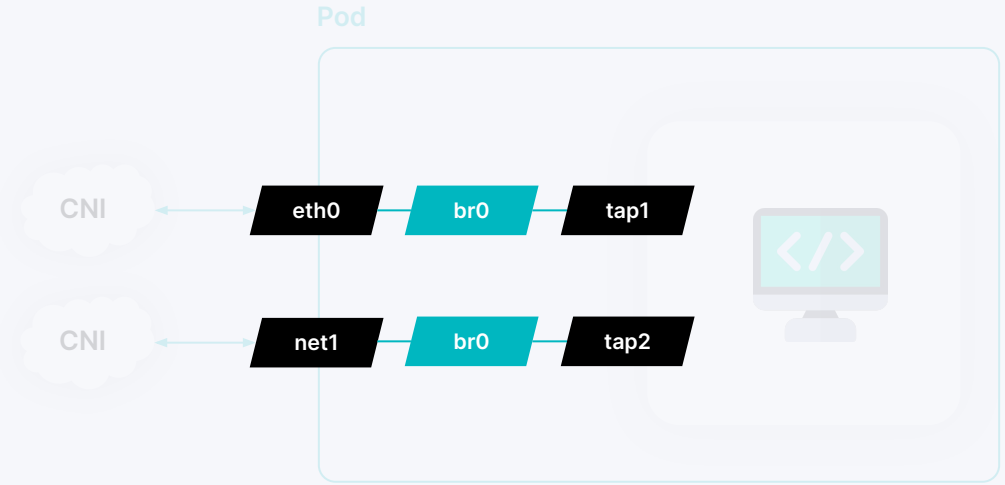
Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

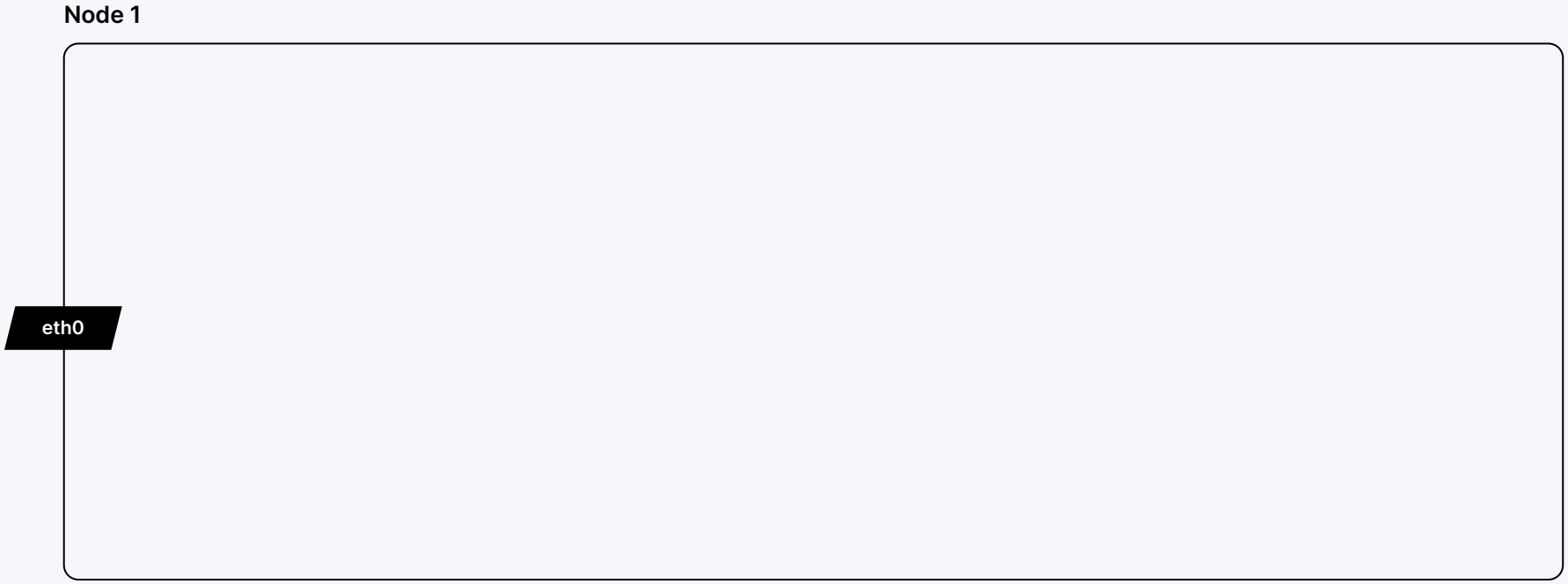
bridge

Connect using a linux bridge

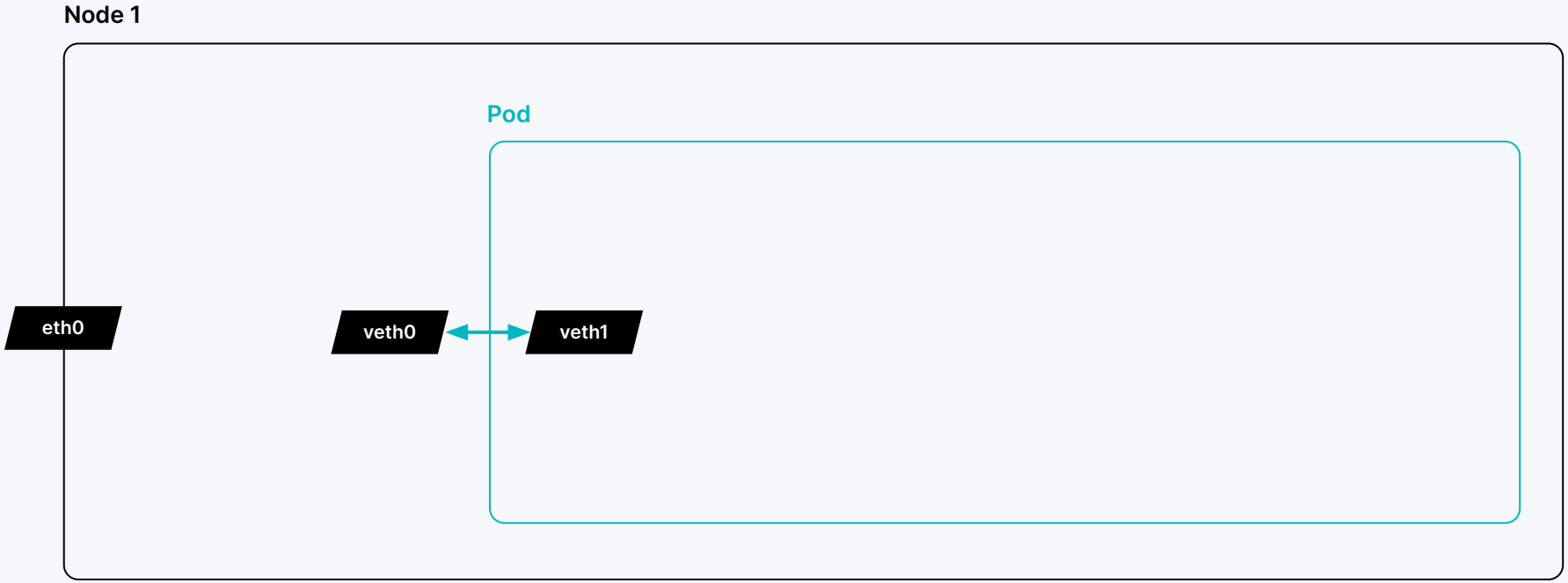


bridge

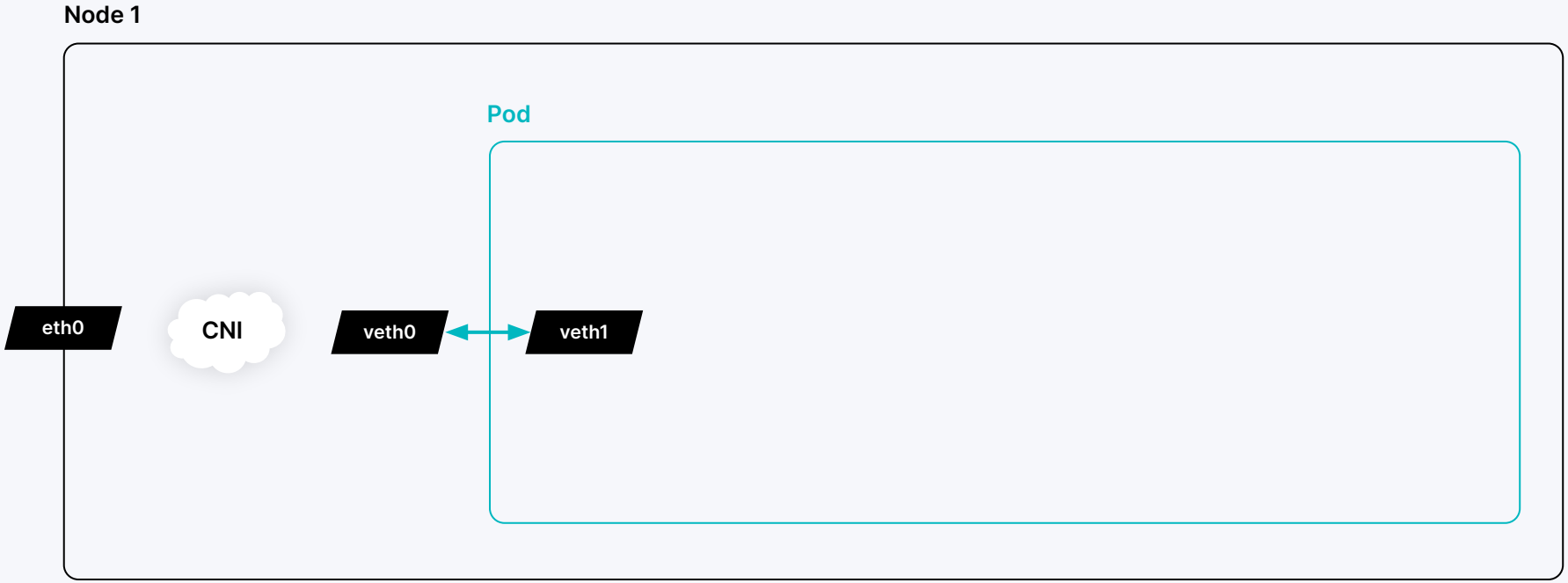
bridge



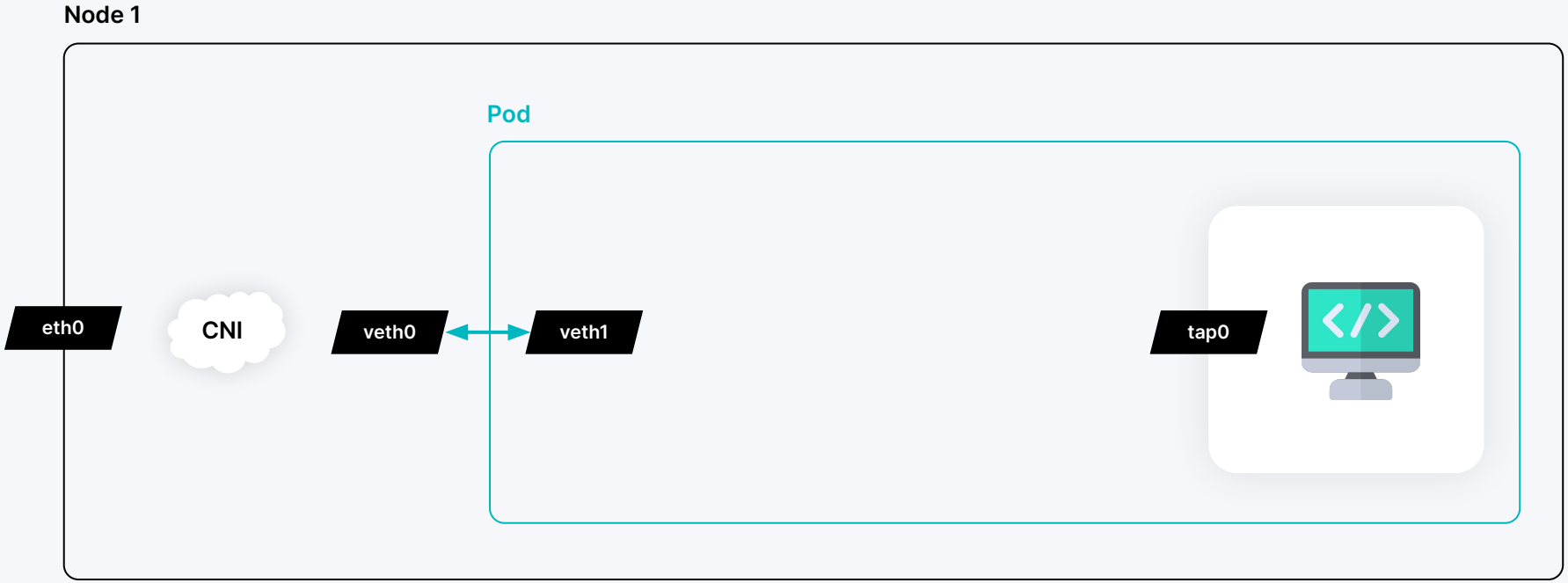
bridge



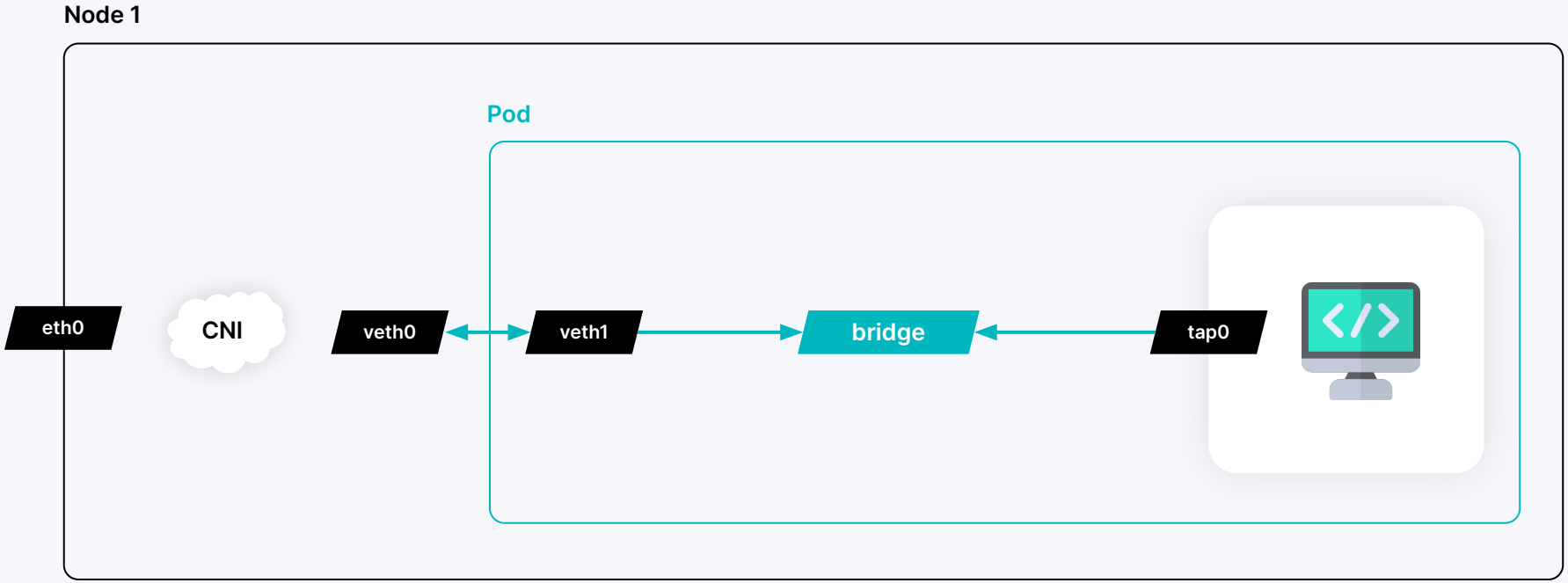
bridge



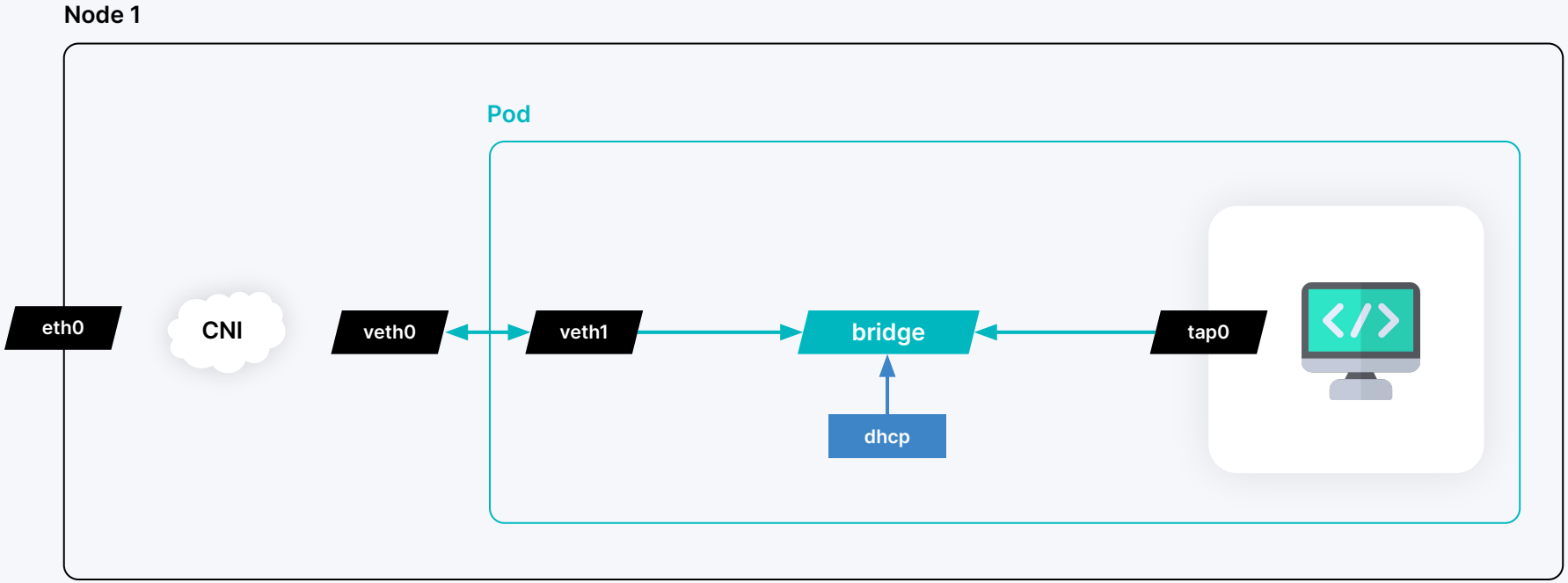
bridge



bridge



bridge



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

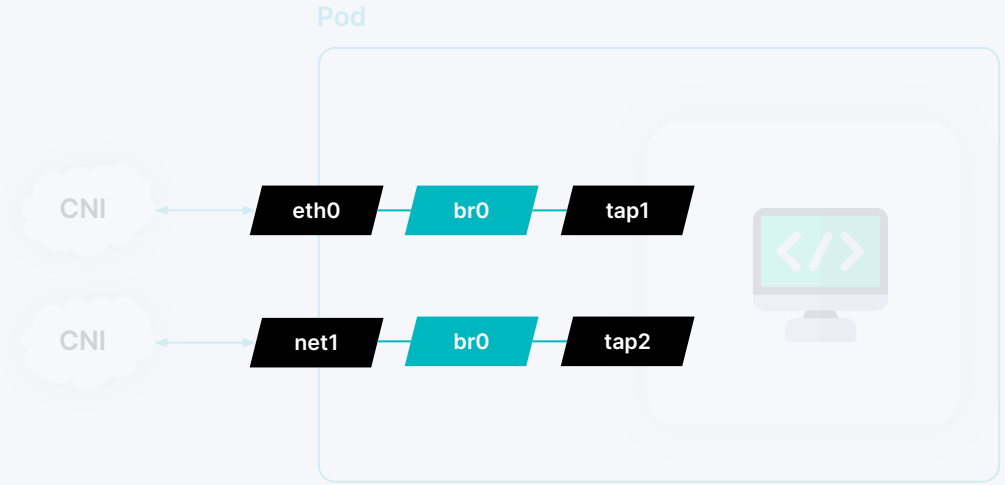
Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

Connect using a linux bridge



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

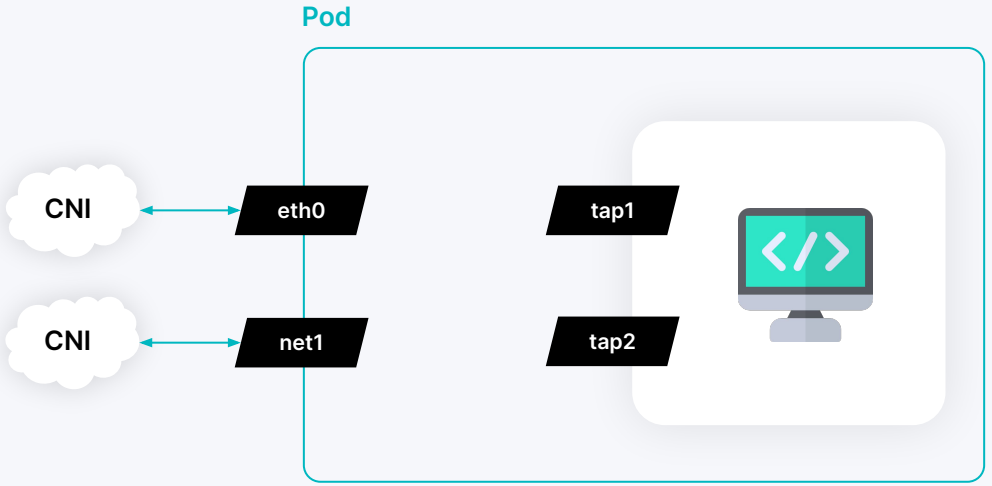
Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

Connect using a linux bridge



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

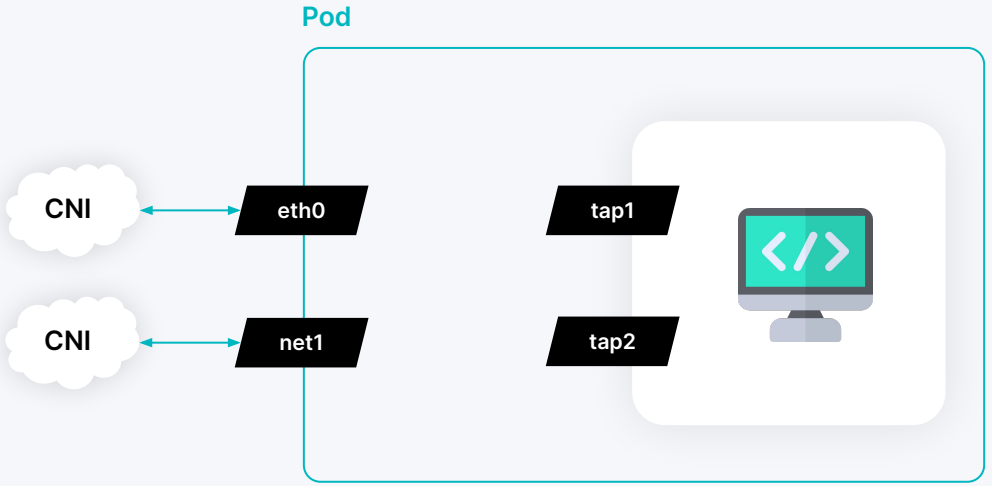
Connect using QEMU user networking mode

bridge

Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

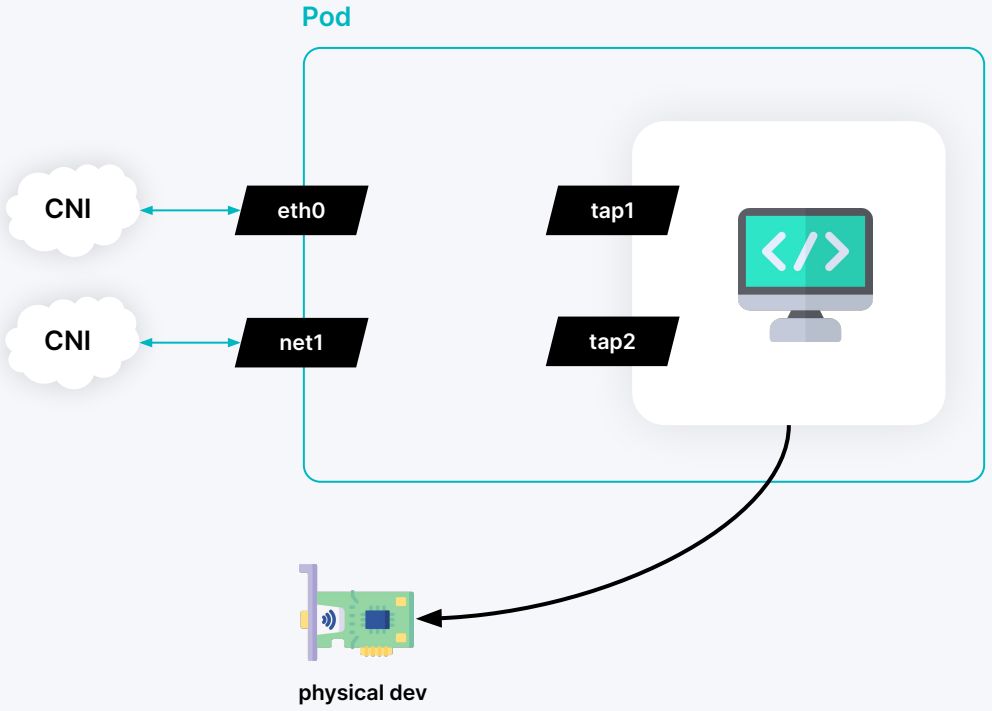
Connect using QEMU user networking mode

bridge

Connect using a linux bridge

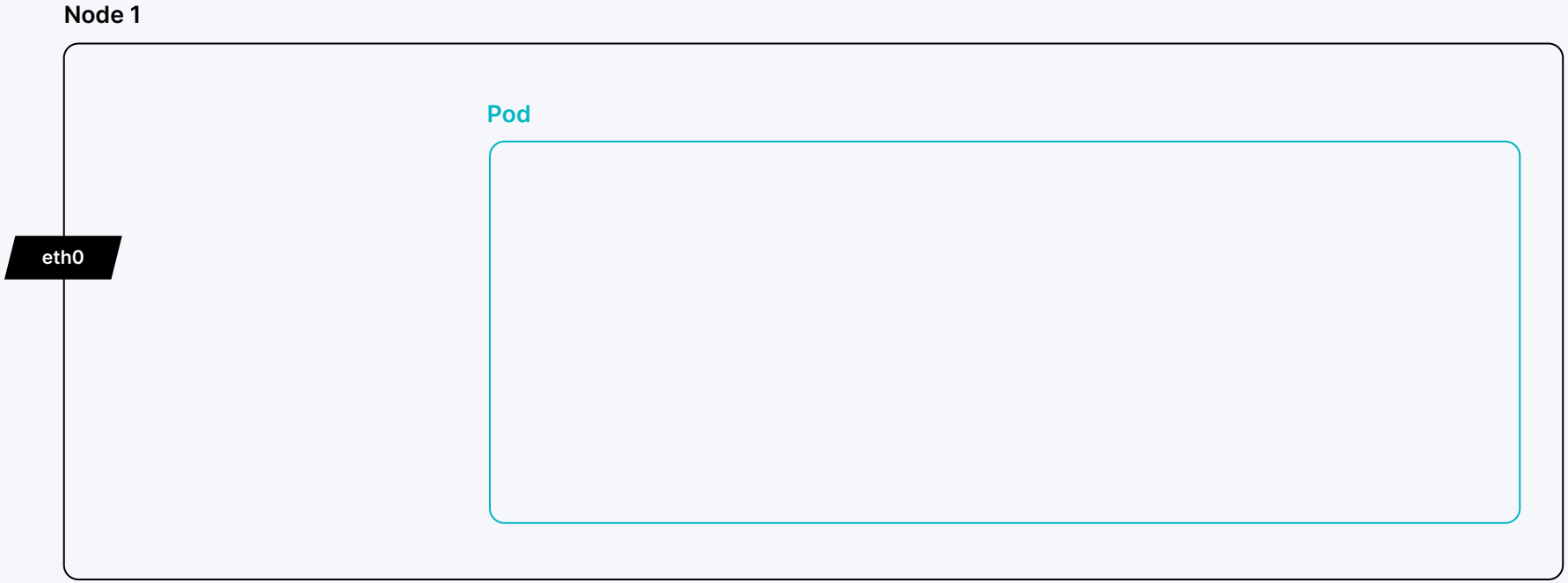
sriov

Pass through a SR-IOV PCI device via vfio

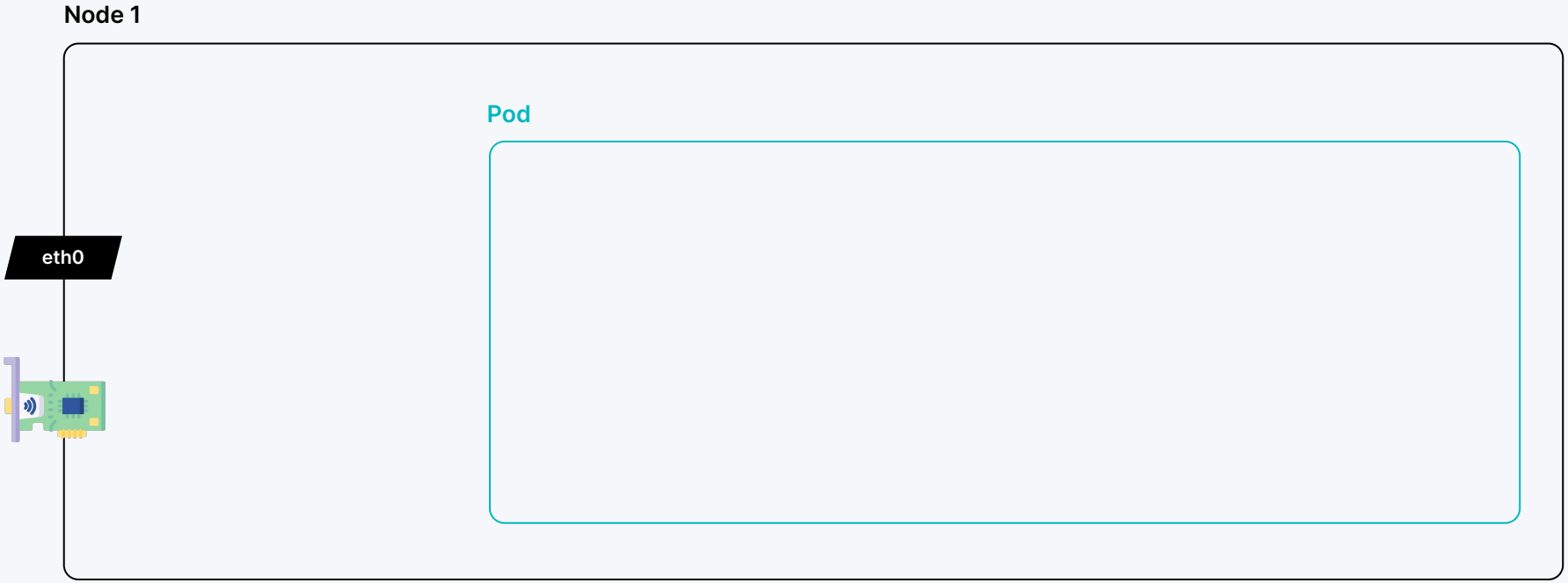


sriov

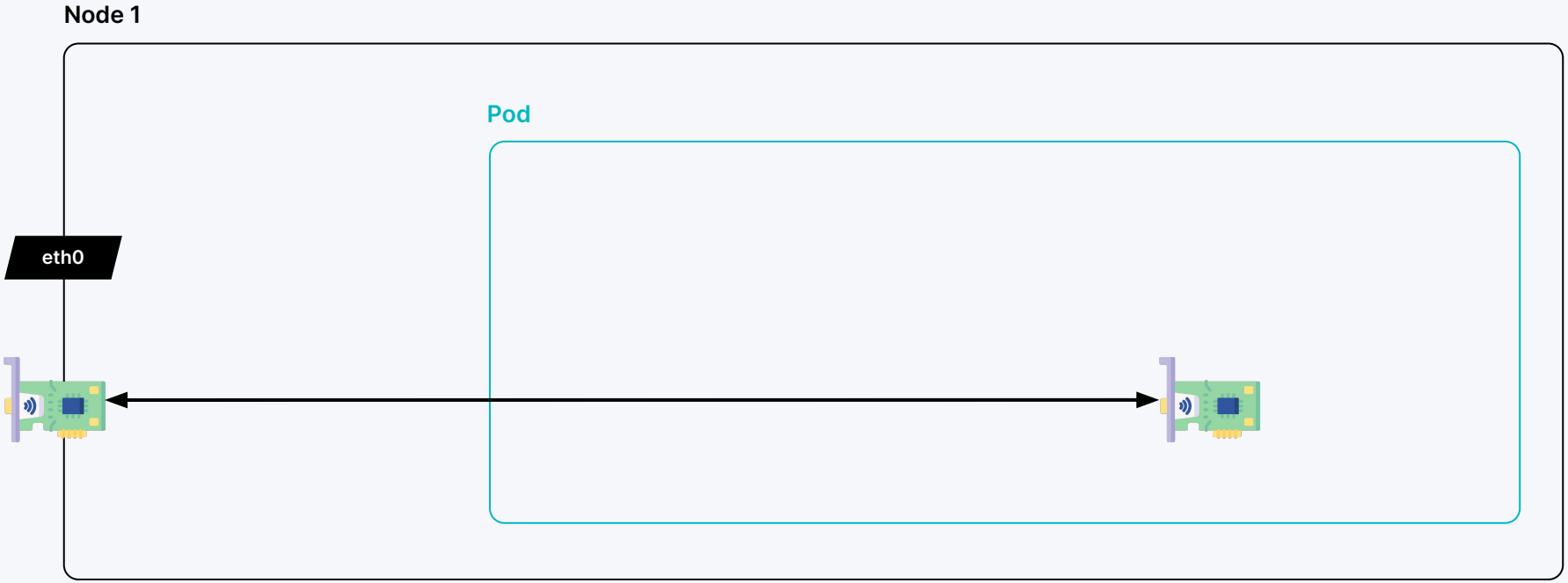
sriov



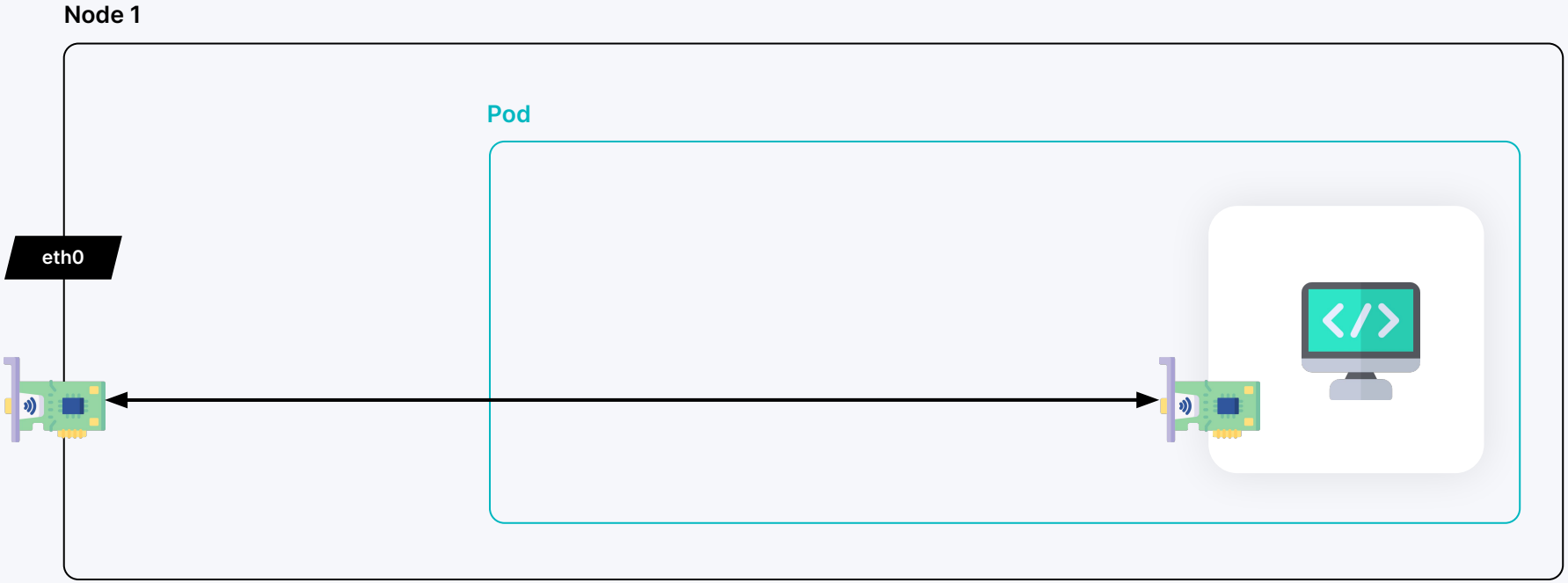
sriov



sriov



sriov



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

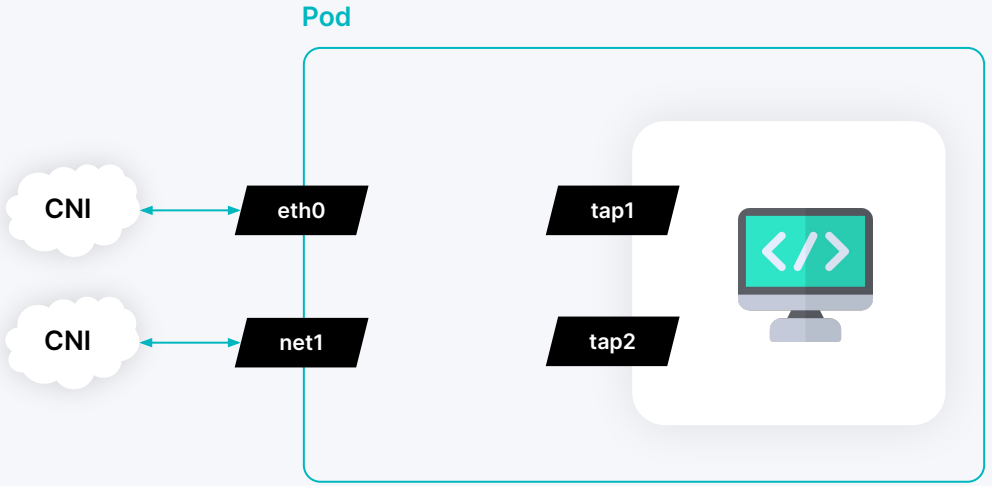
Connect using QEMU user networking mode

bridge

Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

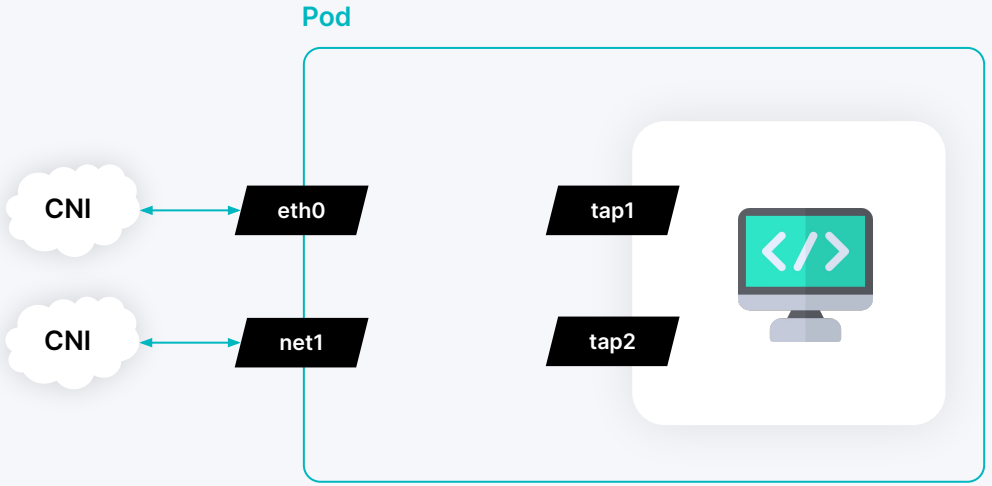
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

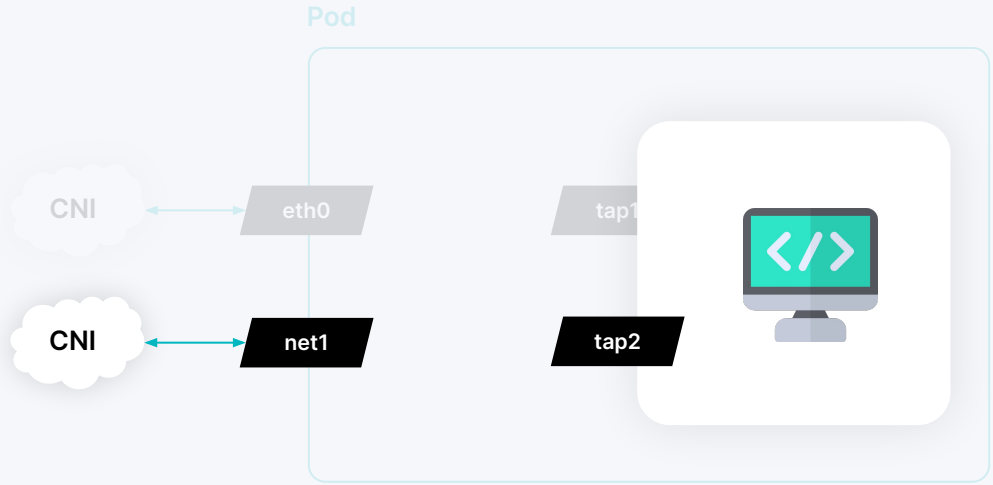
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

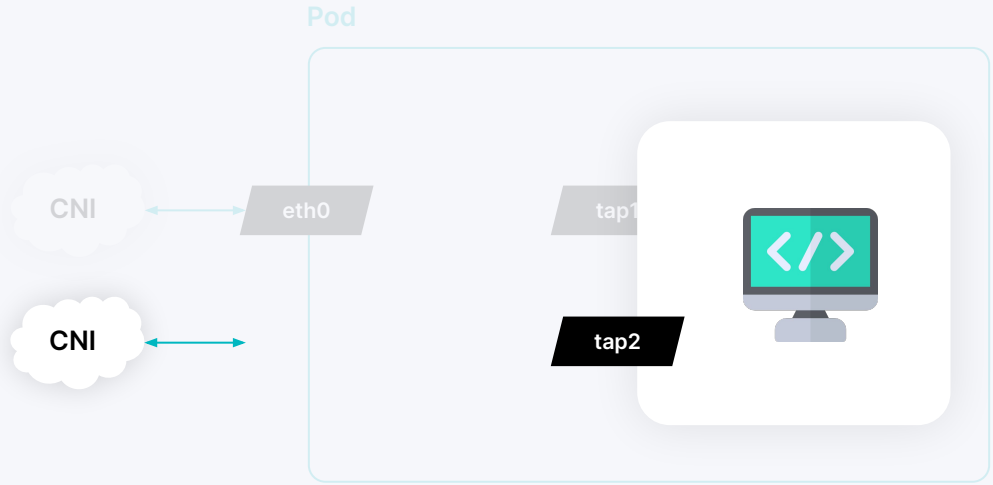
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

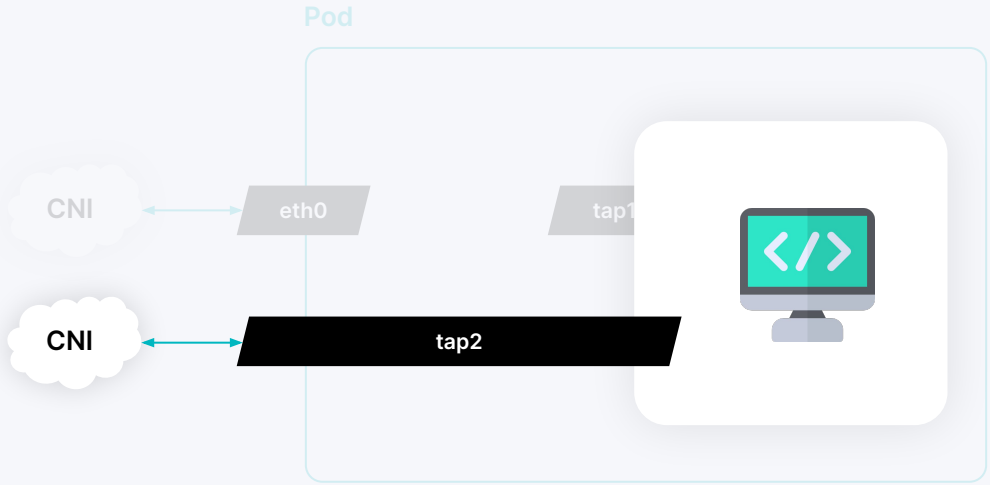
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

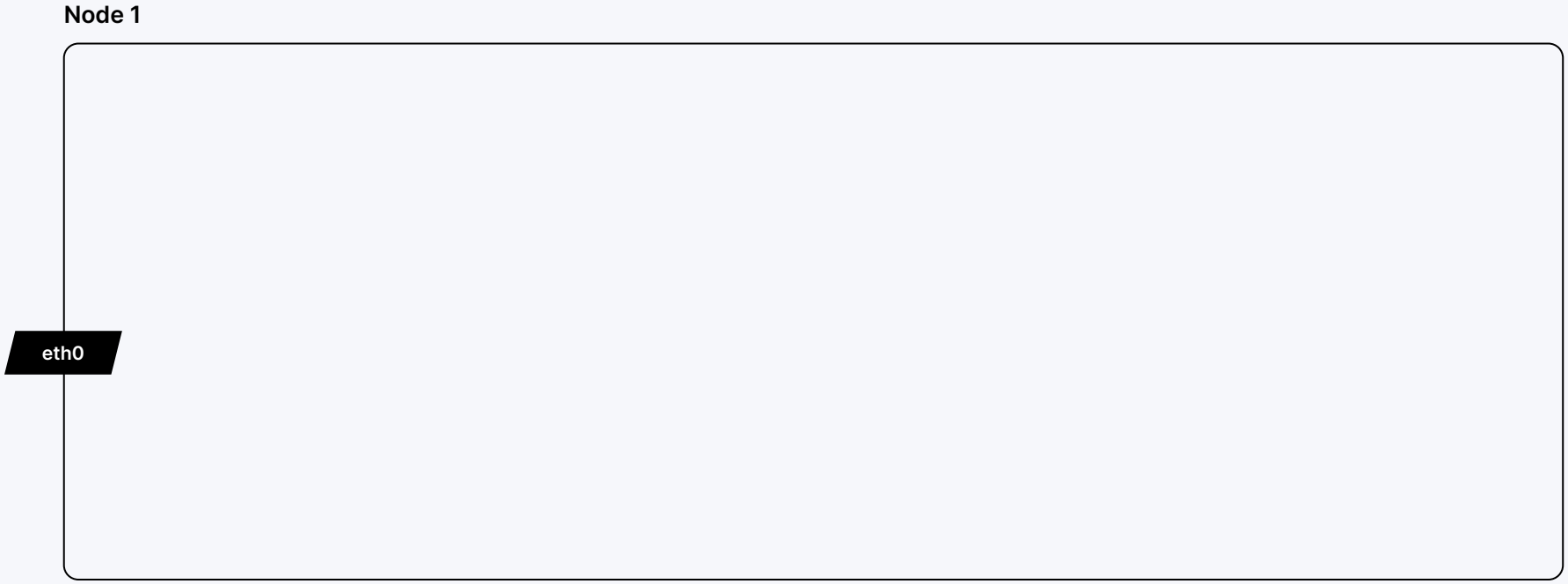
macvtap

Expose directly to the nodes L2 interface

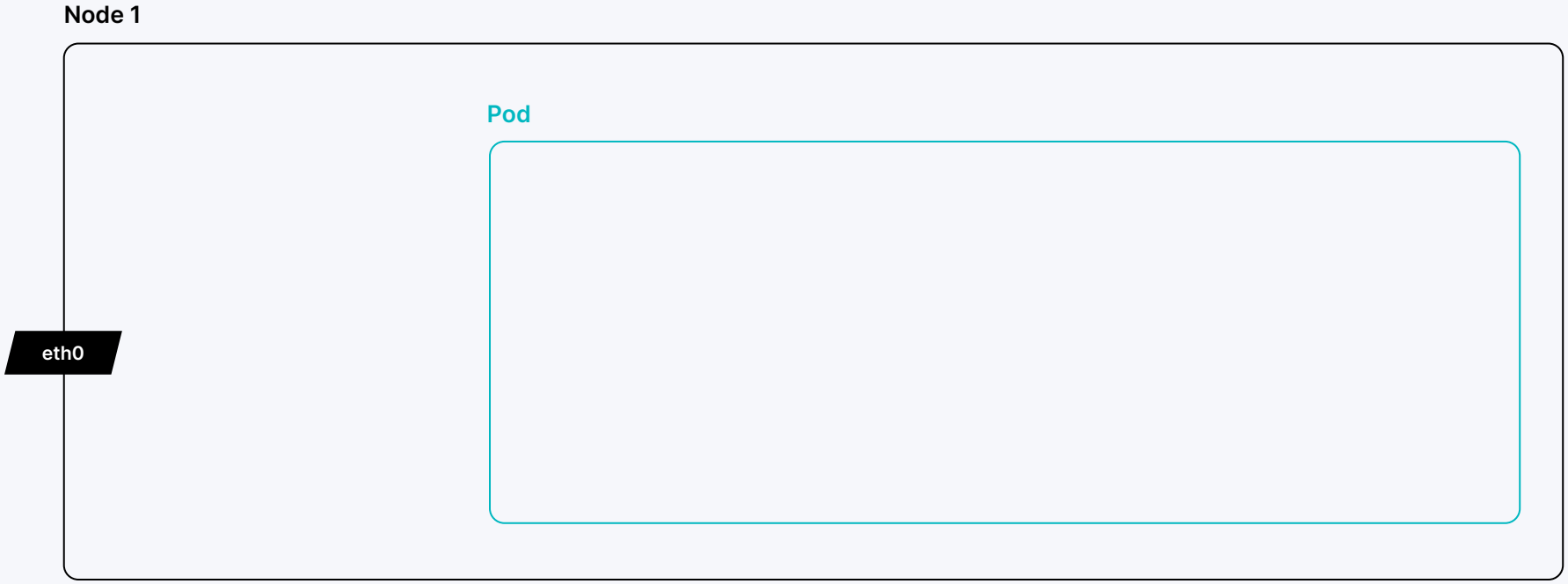


macvtap-cni

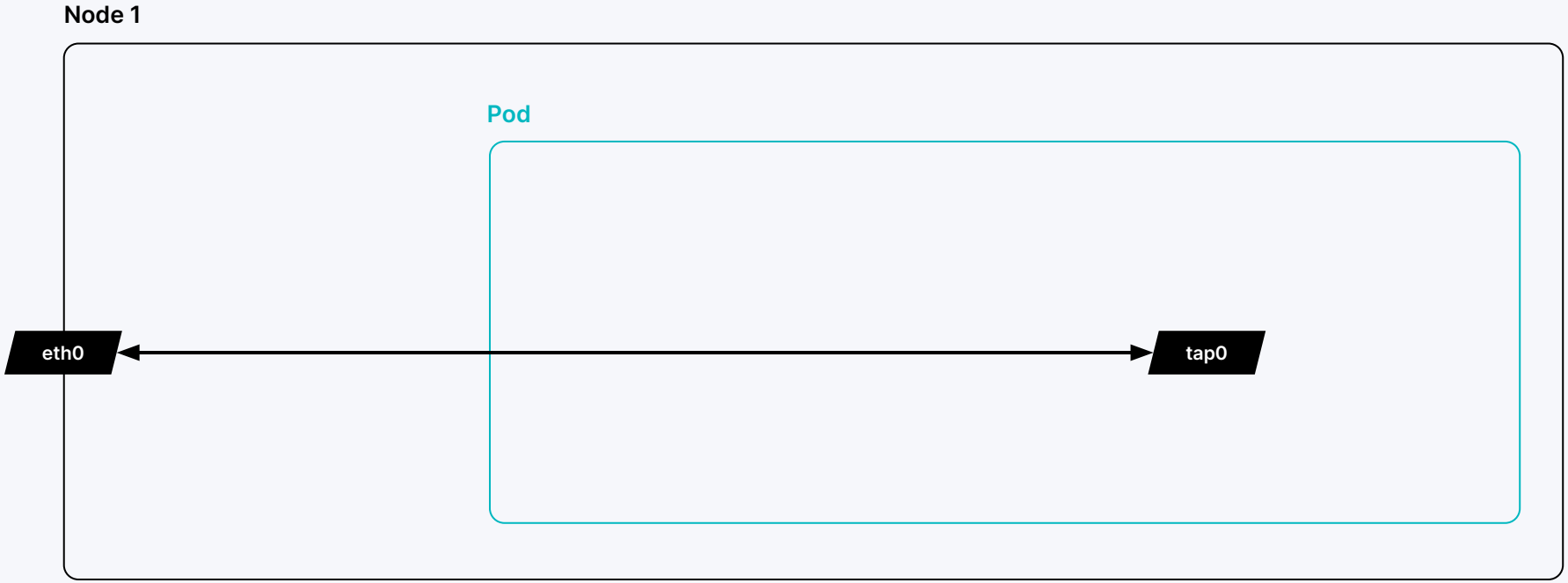
macvtap-cni



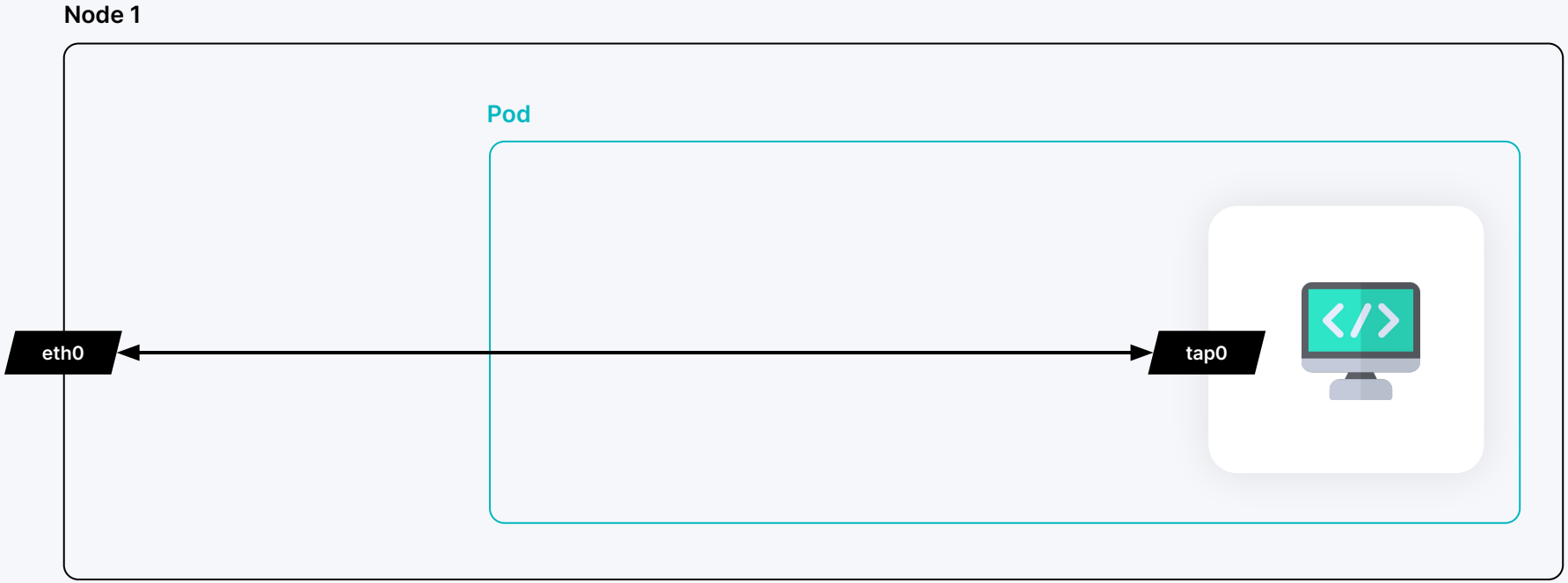
macvtap-cni



macvtap-cni



macvtap-cni



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

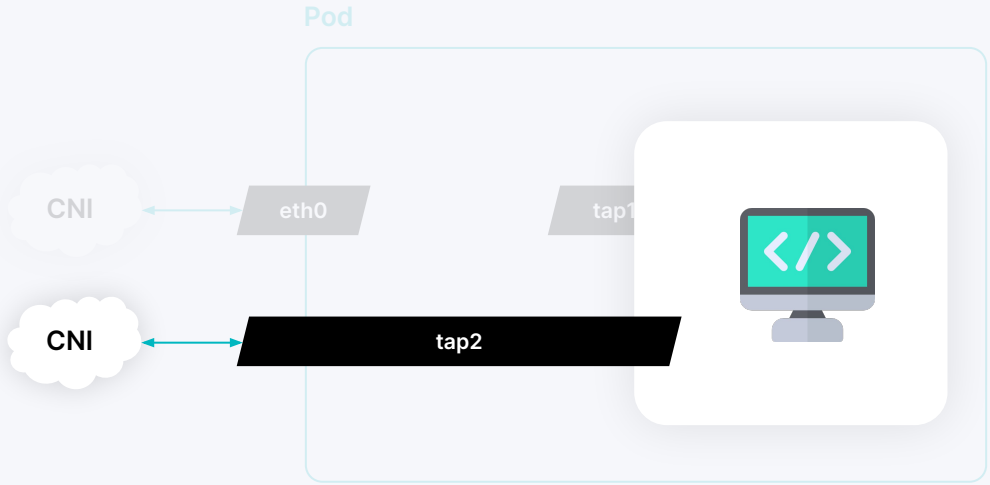
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

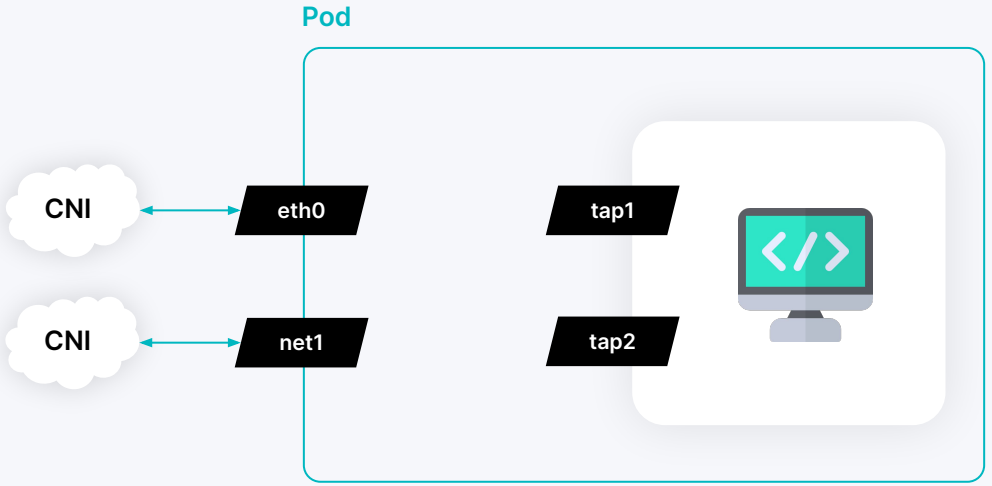
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

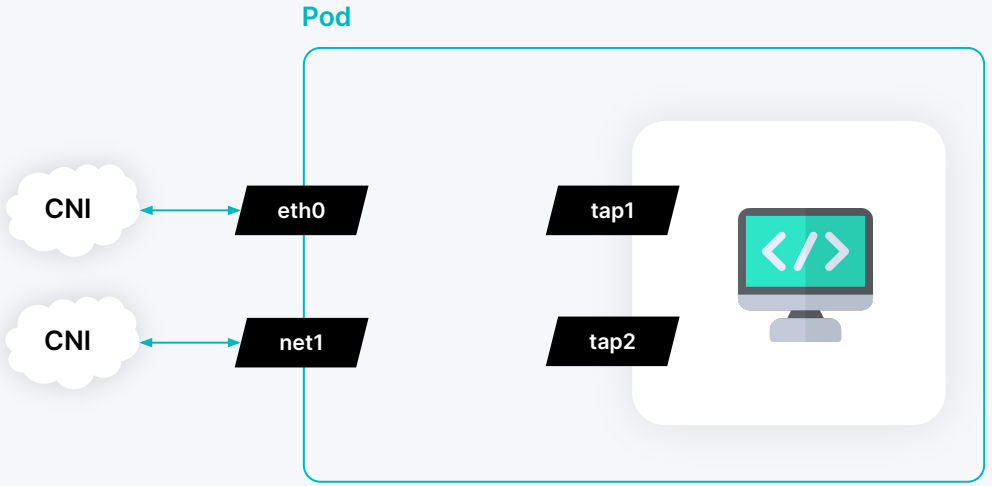
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

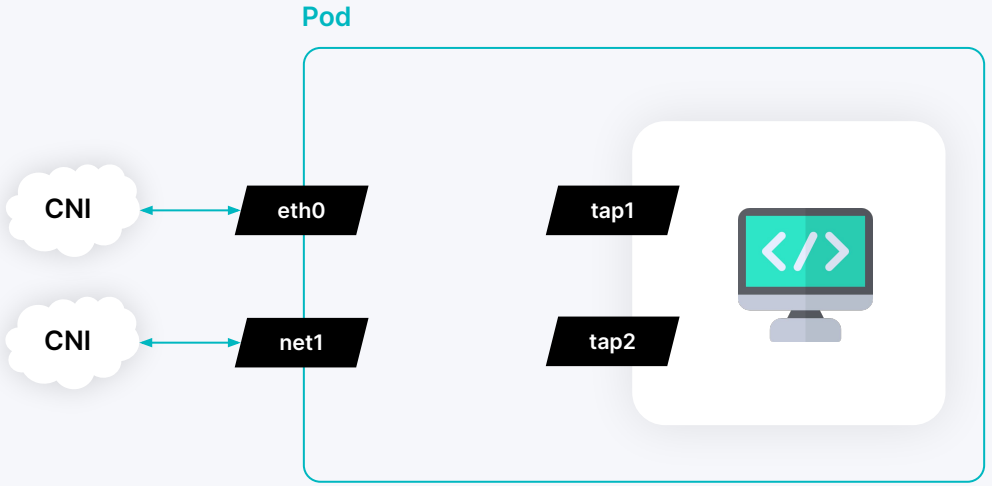
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

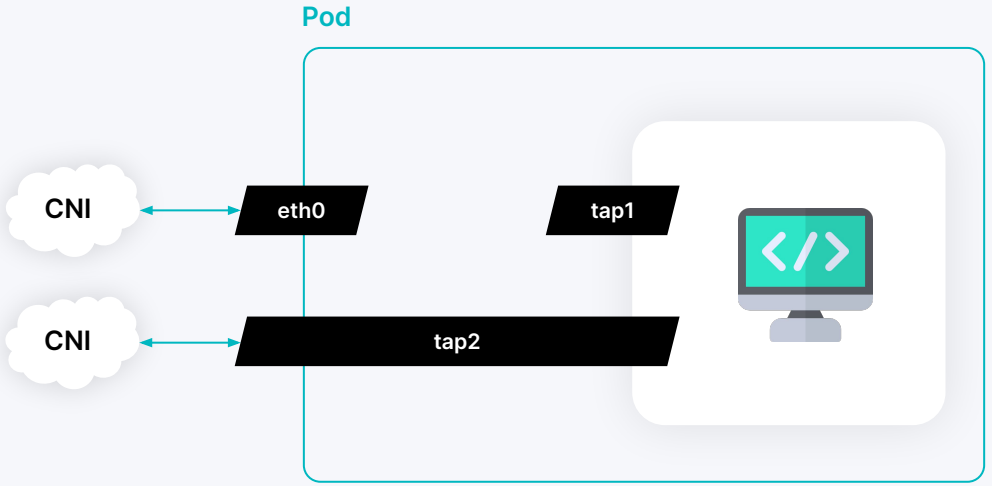
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap 👍

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

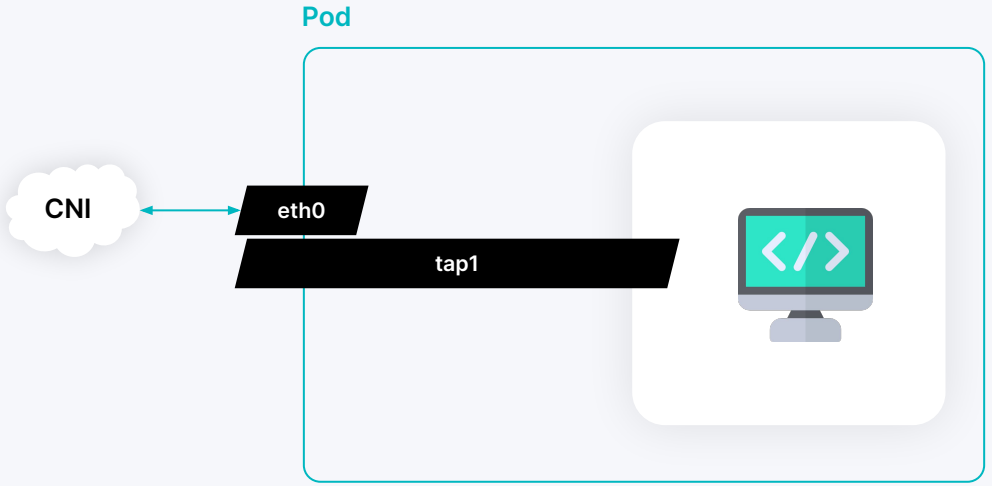
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap **enhanced by us** 👍

Expose directly to the nodes L2 interface

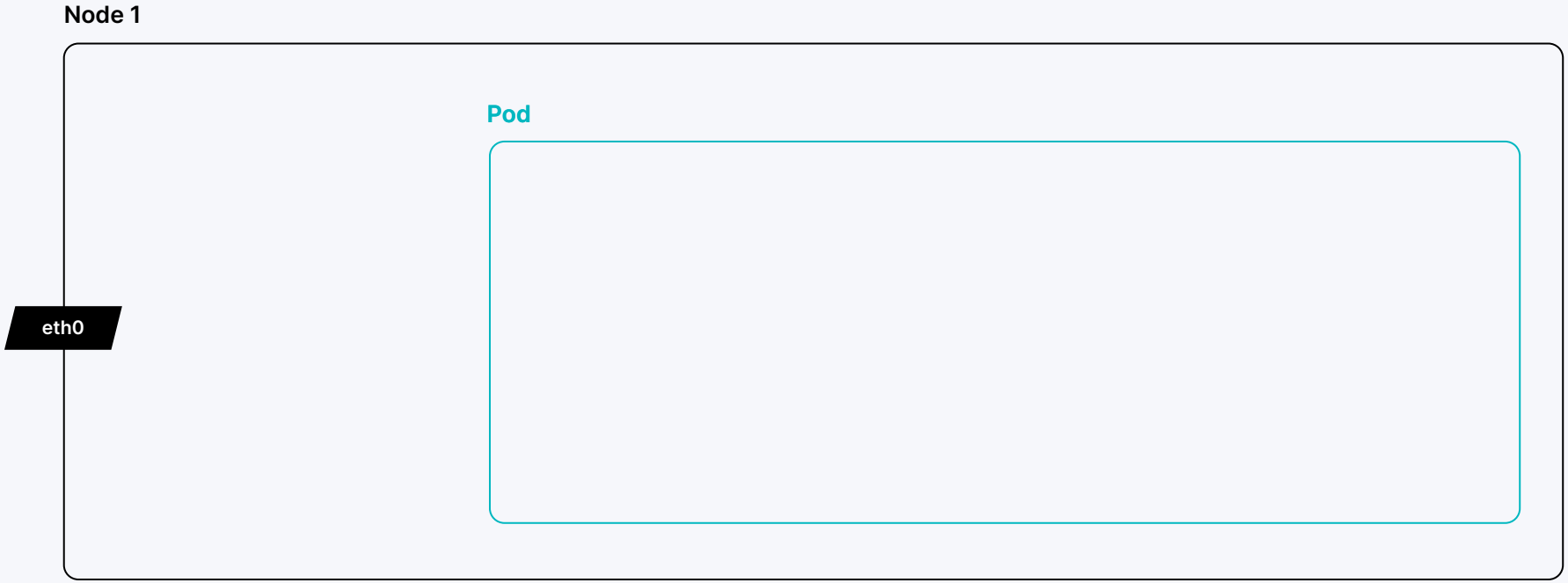


macvtap enhanced by us

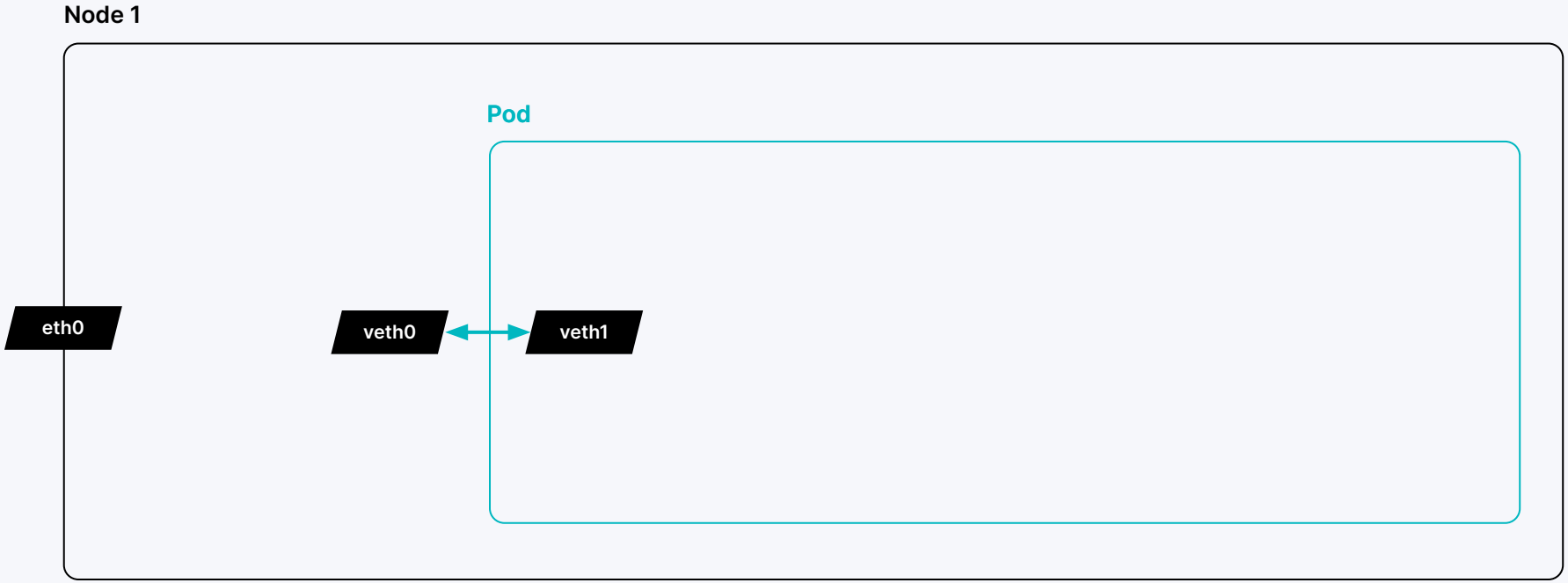
macvtap enhanced by us

Node 1

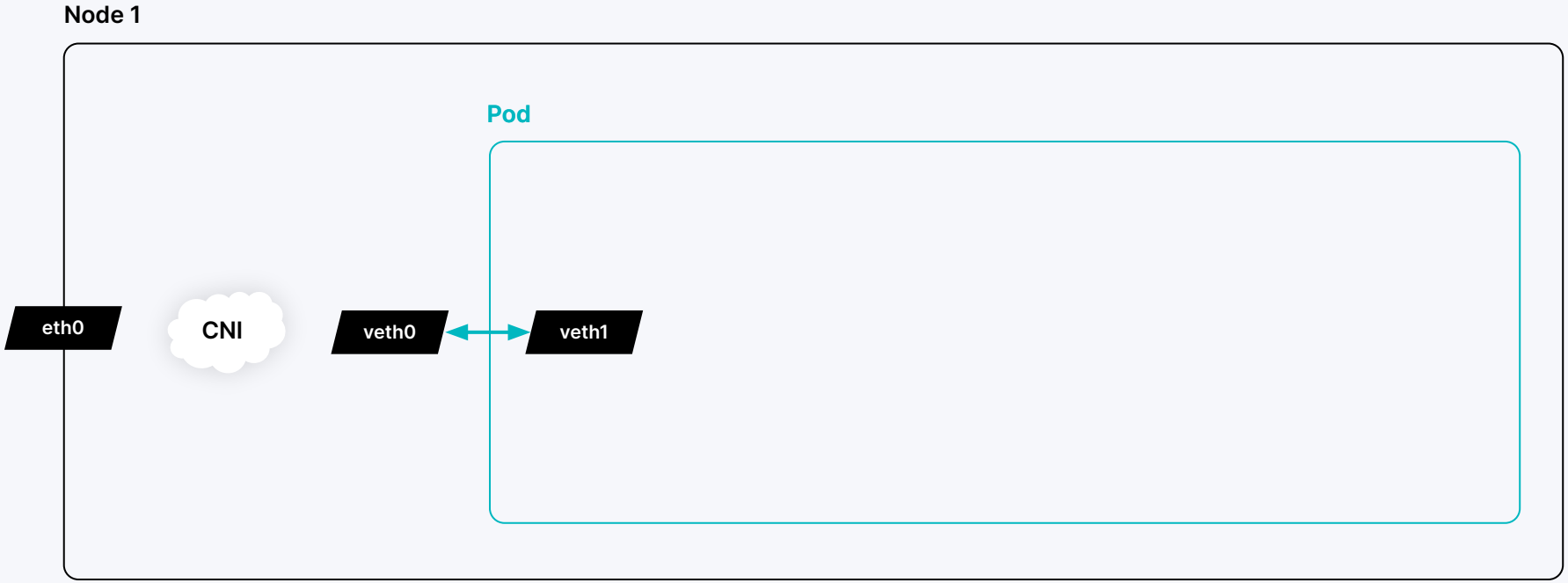
macvtap enhanced by us



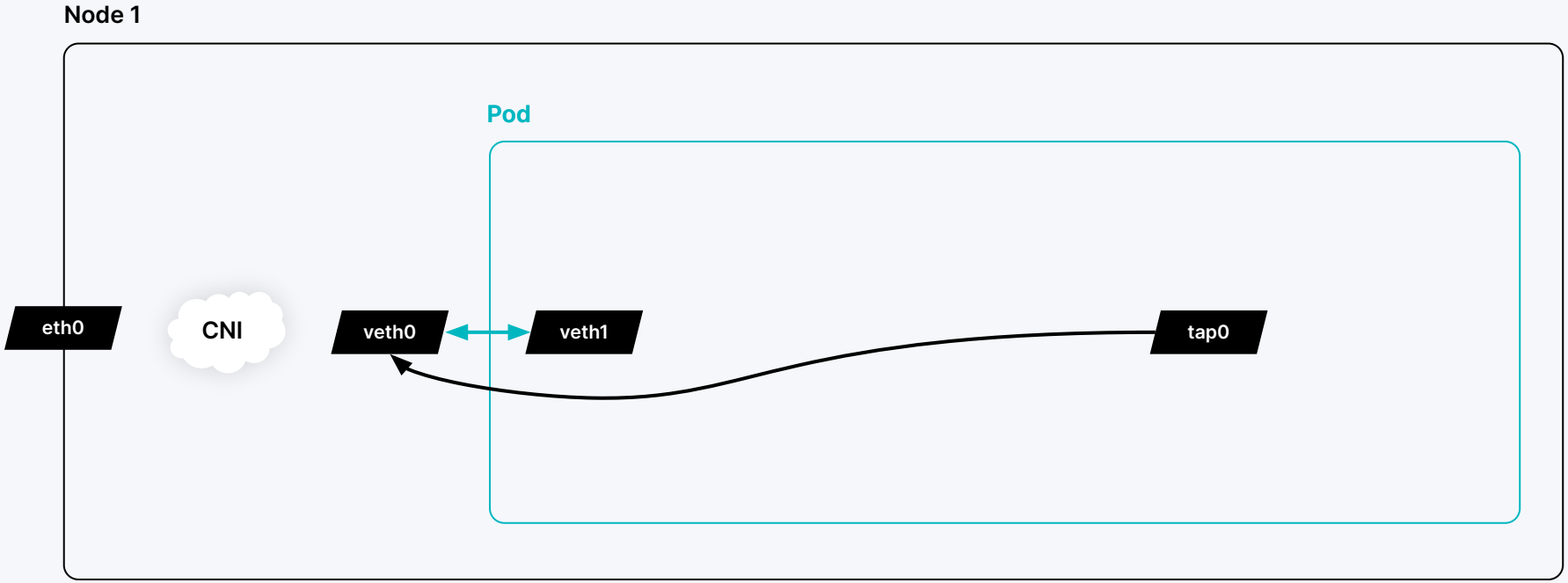
macvtap enhanced by us



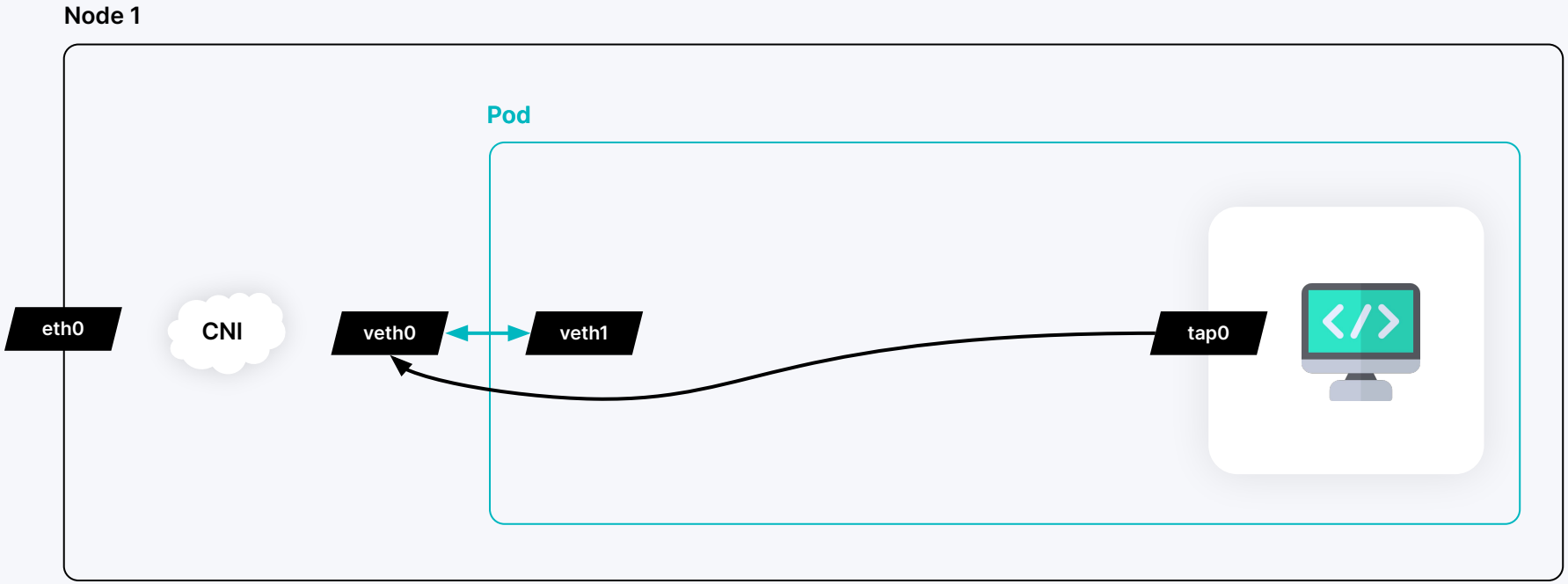
macvtap enhanced by us



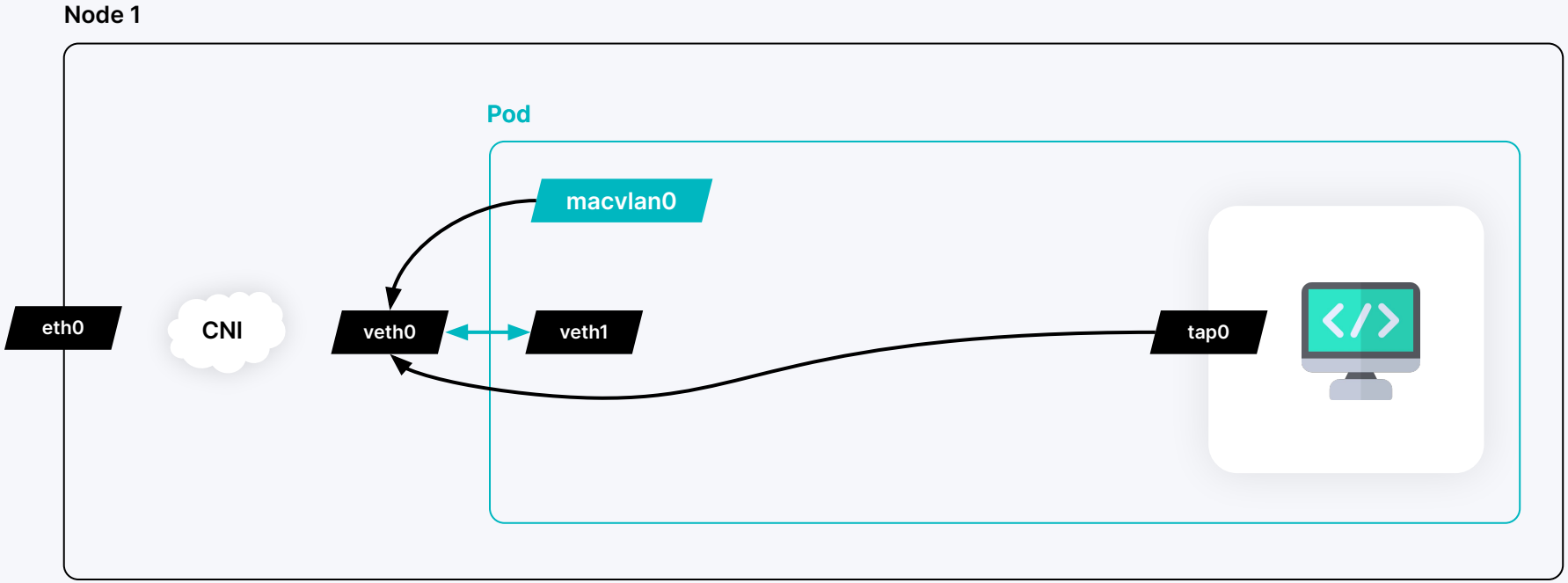
macvtap enhanced by us



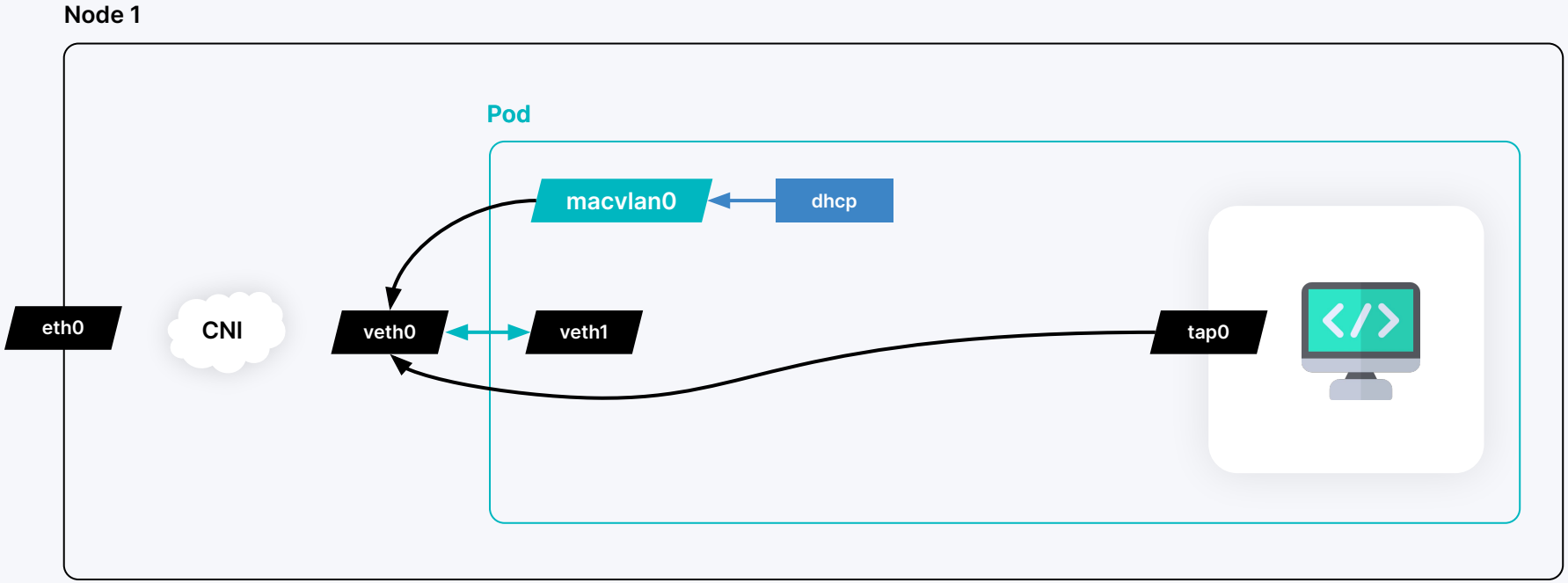
macvtap enhanced by us



macvtap enhanced by us



macvtap enhanced by us



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

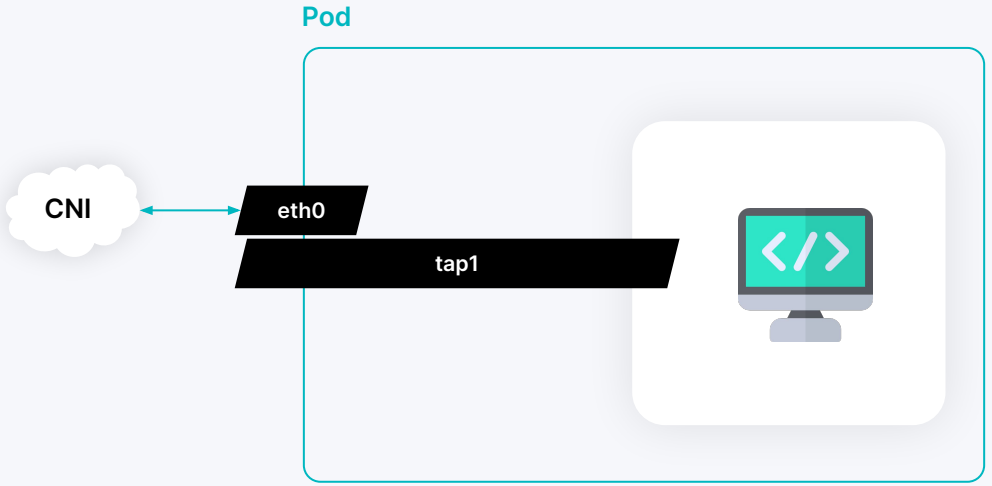
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap **enhanced by us** 👍

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap **enhanced by us** 👍

Expose directly to the nodes L2 interface

Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

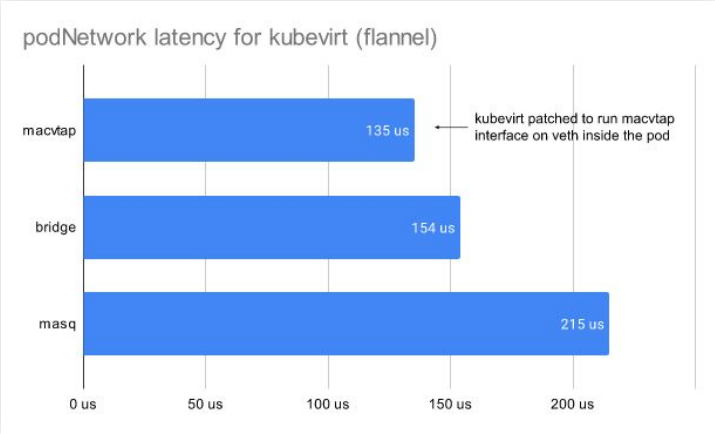
Connect using a linux bridge

sriov

Pass through a SR-IOV PCI device via vfio

macvtap **enhanced by us** 👍

Expose directly to the nodes L2 interface



Kubernetes networking

Backend

pod

Default Kubernetes network

multus

Secondary network provided using Multus

Frontend

masquerade

Connect using Iptables rules to nat the traffic

slirp

Connect using QEMU user networking mode

bridge

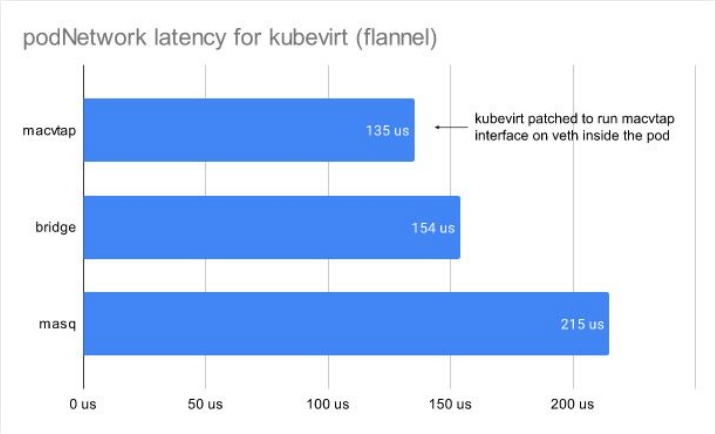
Connect using a linux bridge

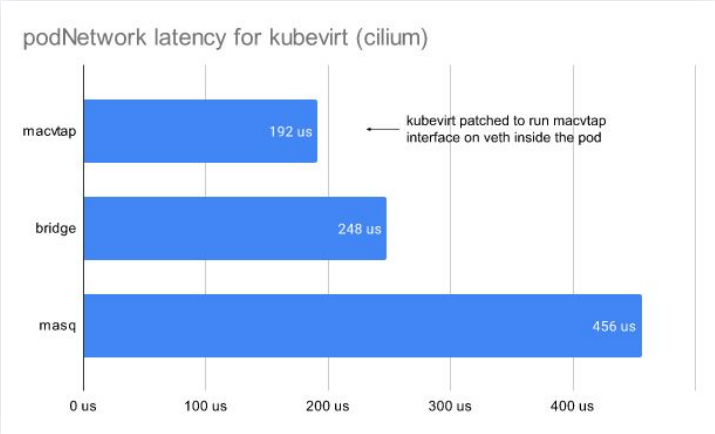
sriov

Pass through a SR-IOV PCI device via vfio

macvtap **enhanced by us** 👍

Expose directly to the nodes L2 interface

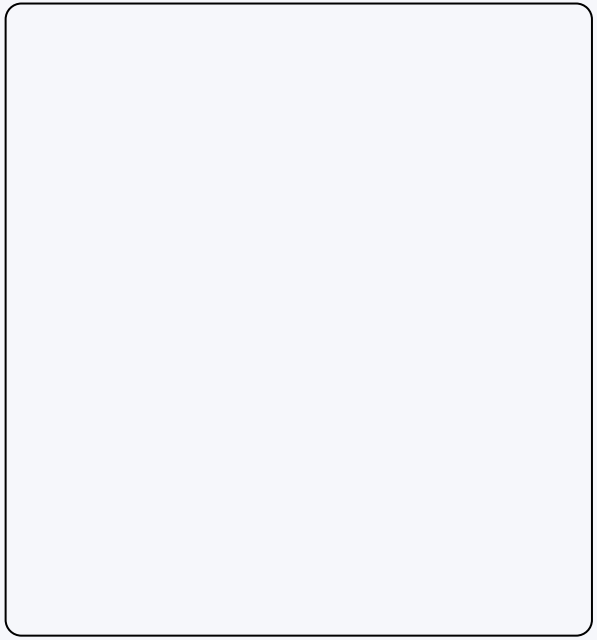




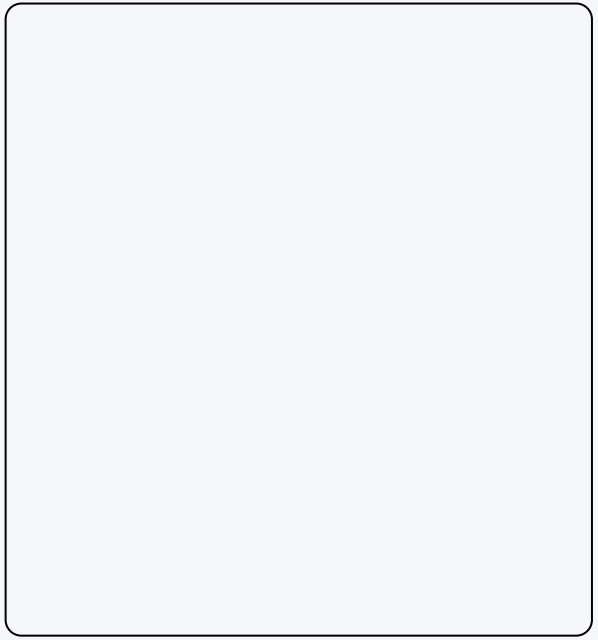
How to make static IPs

How to make static IPs

Node 1



Node 2



How to make static IPs

Node 1

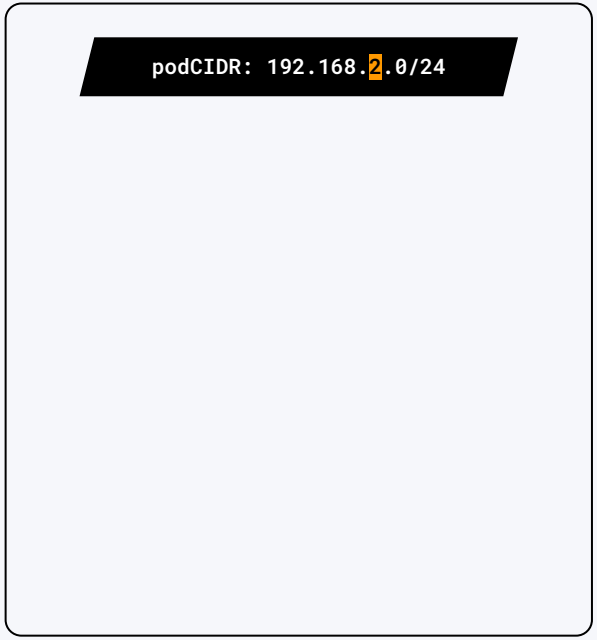
```
podCIDR: 192.168.1.0/24
```

Node 2

How to make static IPs

Node 1

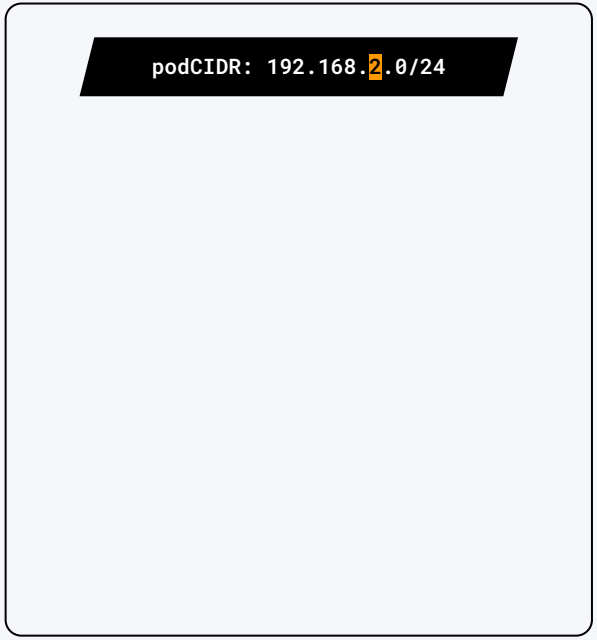
Node 2



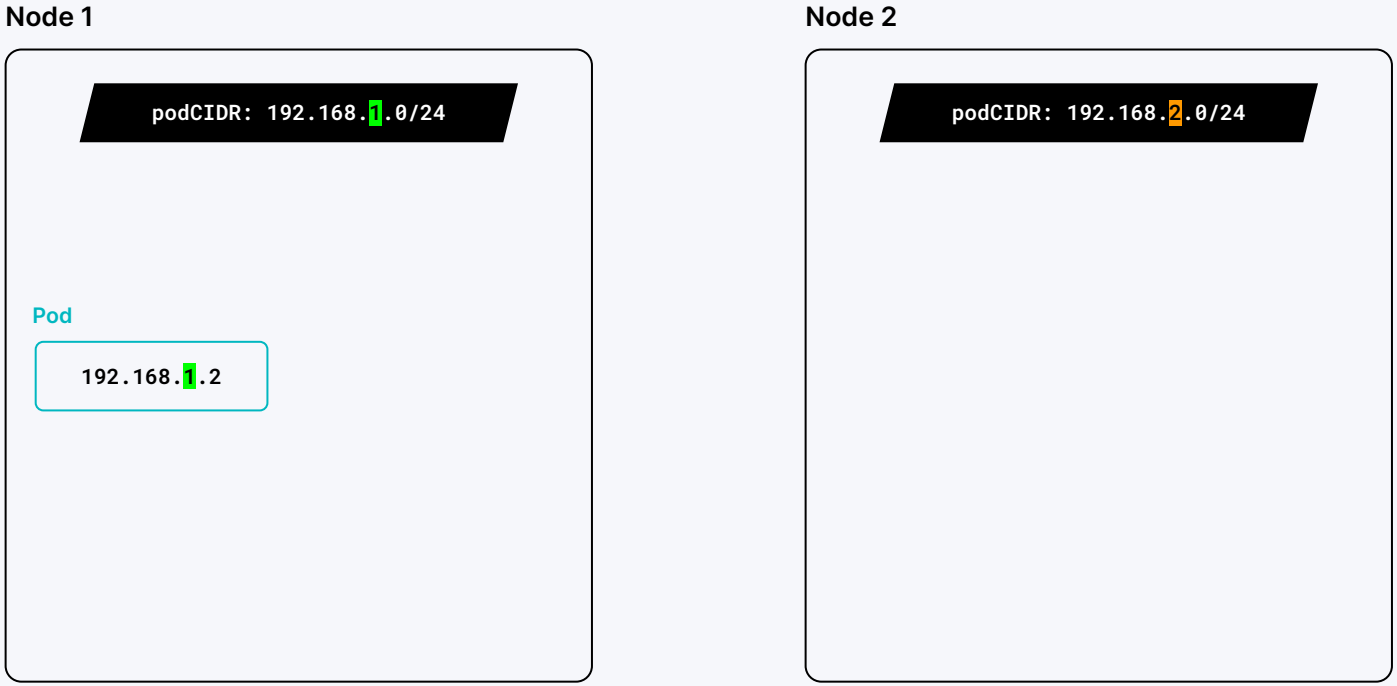
How to make static IPs

Node 1

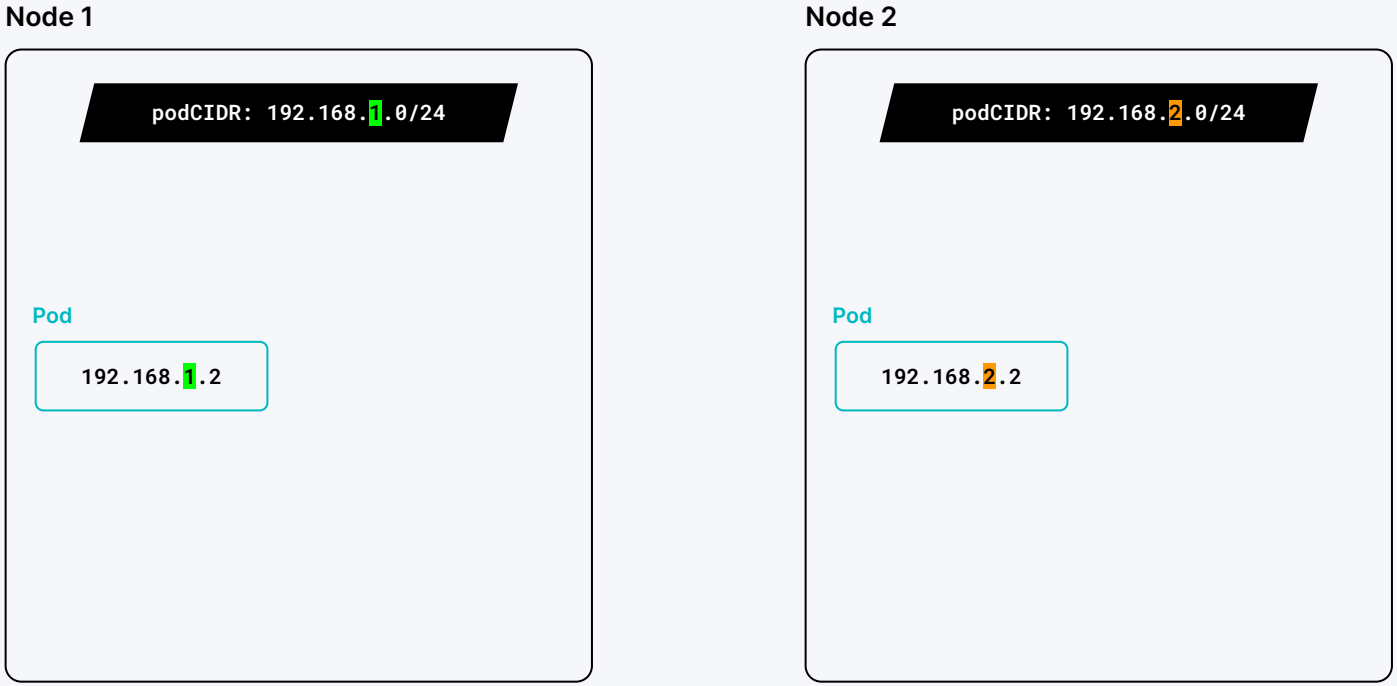
Node 2



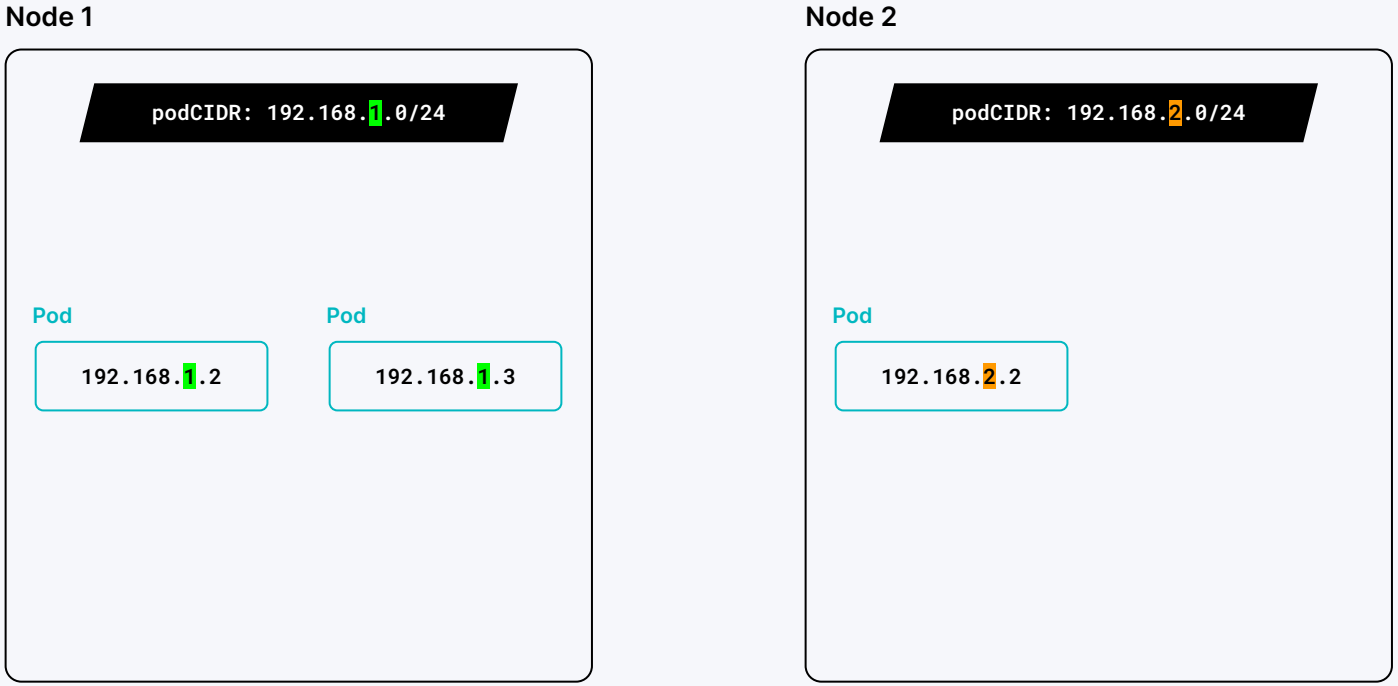
How to make static IPs



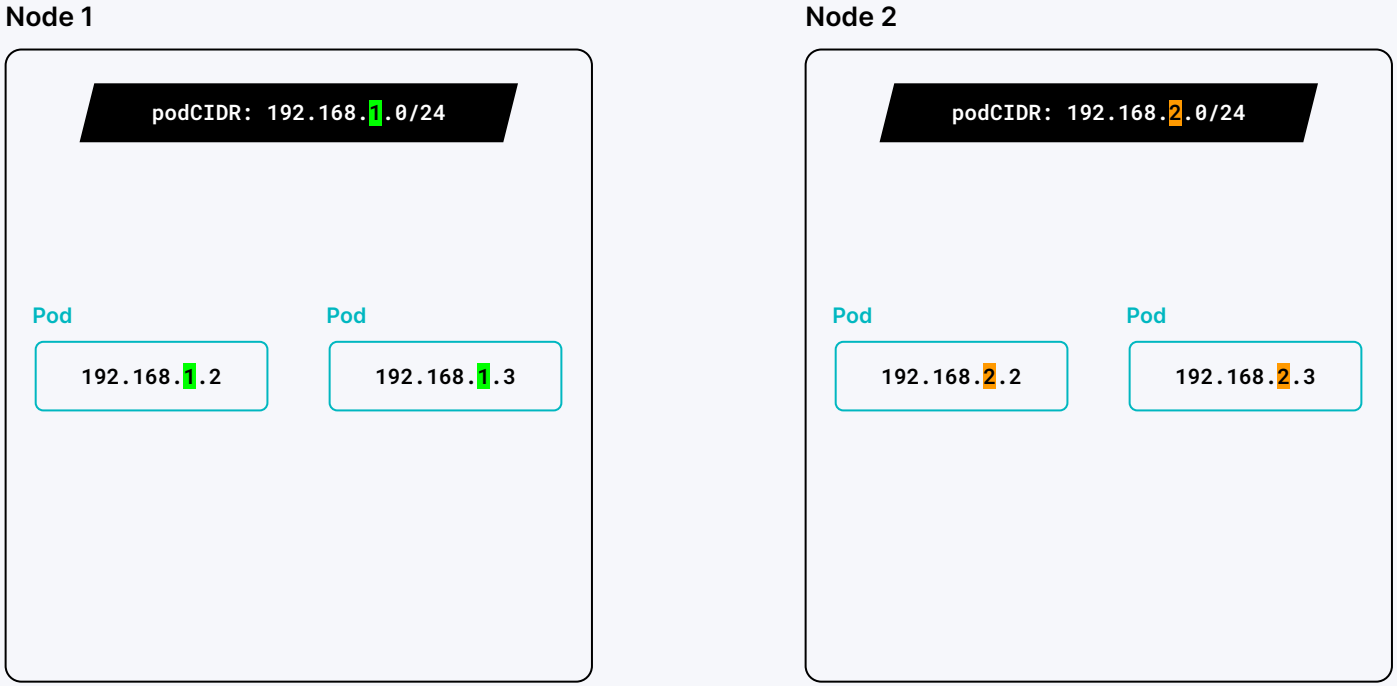
How to make static IPs



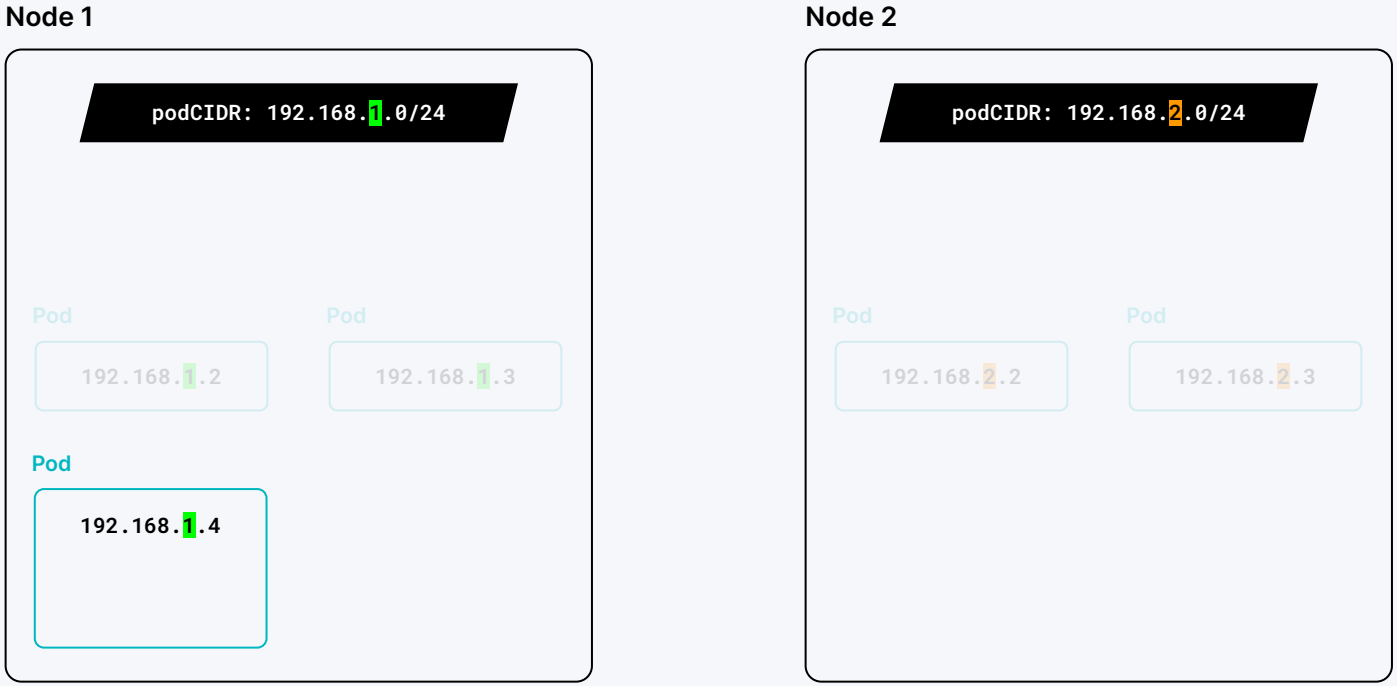
How to make static IPs



How to make static IPs



How to make static IPs



How to make static IPs

Node 1

podCIDR: 192.168.1.0/24

Pod

192.168.1.2

Pod

192.168.1.3

Pod

192.168.1.4

Node 2

podCIDR: 192.168.2.0/24

Pod

192.168.2.2

Pod

192.168.2.3

How to make static IPs

Node 1

podCIDR: 192.168.1.0/24

Pod

192.168.1.2

Pod

192.168.1.3

Pod

192.168.1.4

Node 2

podCIDR: 192.168.2.0/24

Pod

192.168.2.2

Pod

192.168.2.3

Pod

192.168.2.5

How to make static IPs

Node 1

podCIDR: 192.168.1.0/24

Pod

192.168.1.2

Pod

192.168.1.3

Pod

192.168.1.4

Node 2

podCIDR: 192.168.2.0/24

Pod

192.168.2.2

Pod

192.168.2.3

Pod

192.168.2.5

How to make static IPs

Node 1

podCIDR: 192.168.1.0/24

Pod

192.168.1.2

Pod

192.168.1.3

Pod

192.168.1.4

Node 2

podCIDR: 192.168.2.0/24

Pod

192.168.2.2

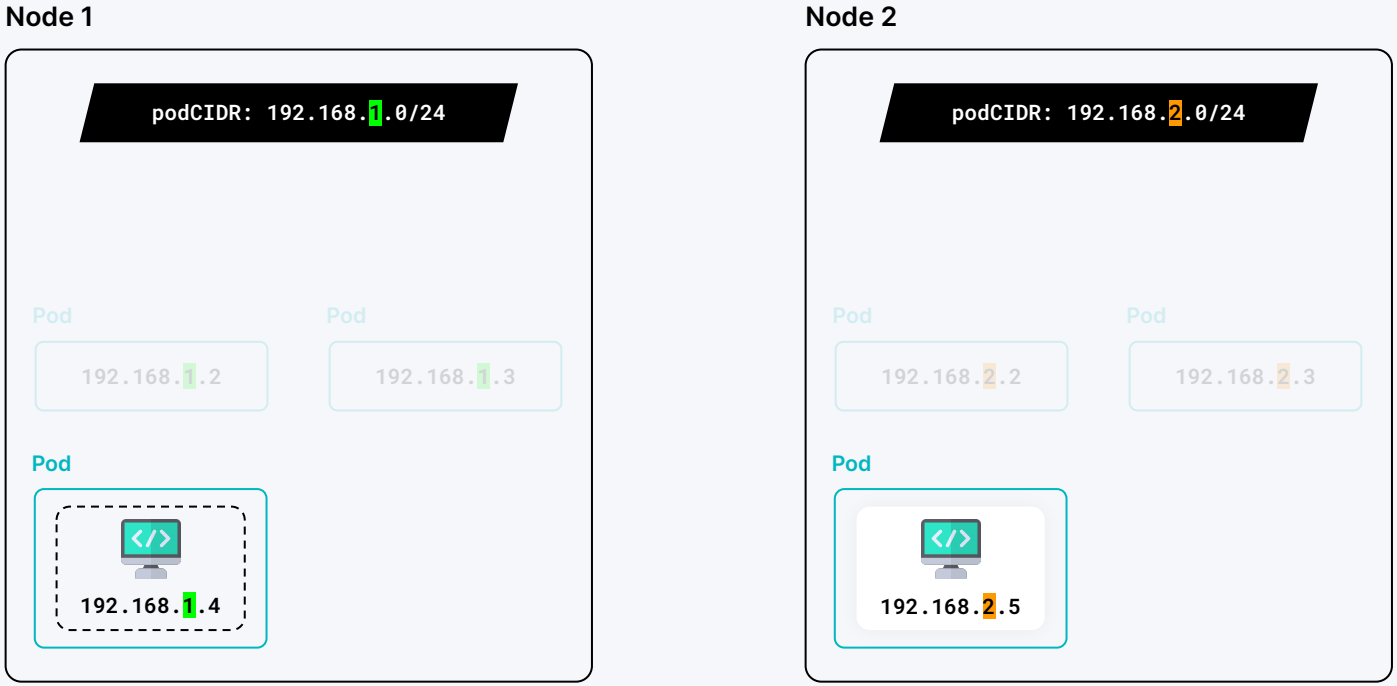
Pod

192.168.2.3

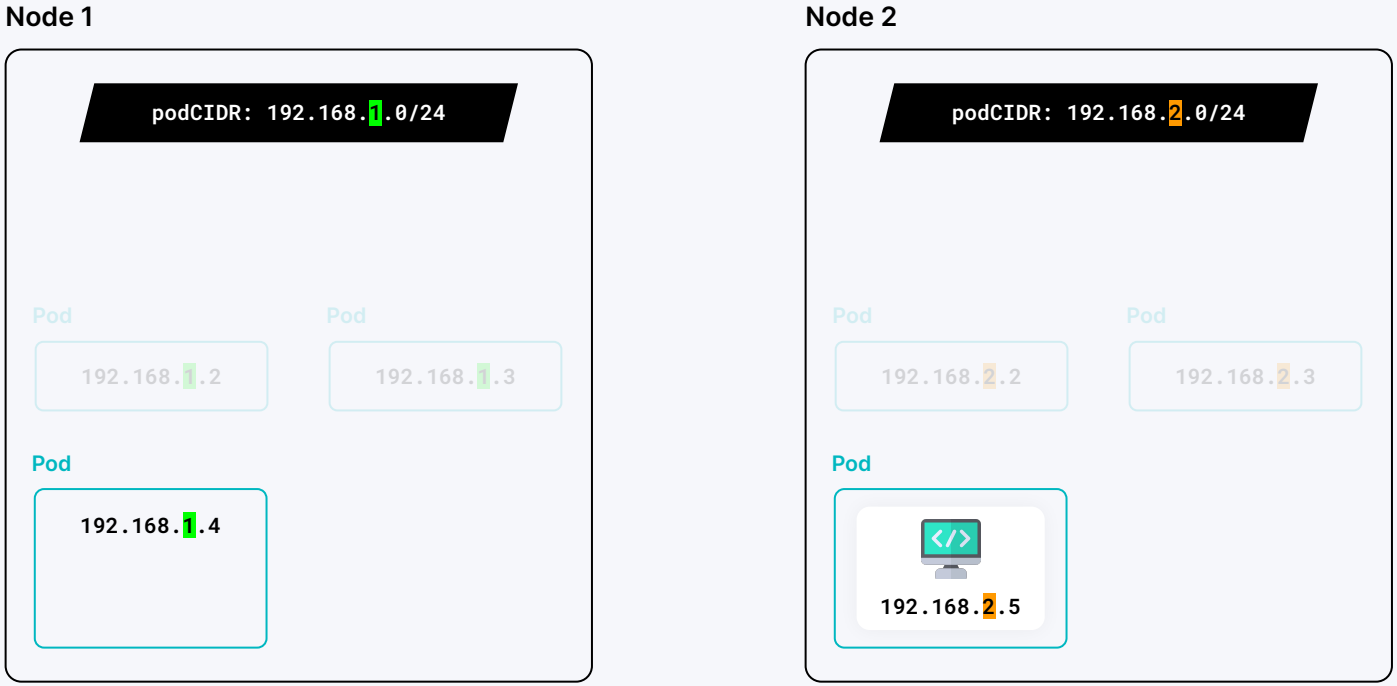
Pod

192.168.2.5

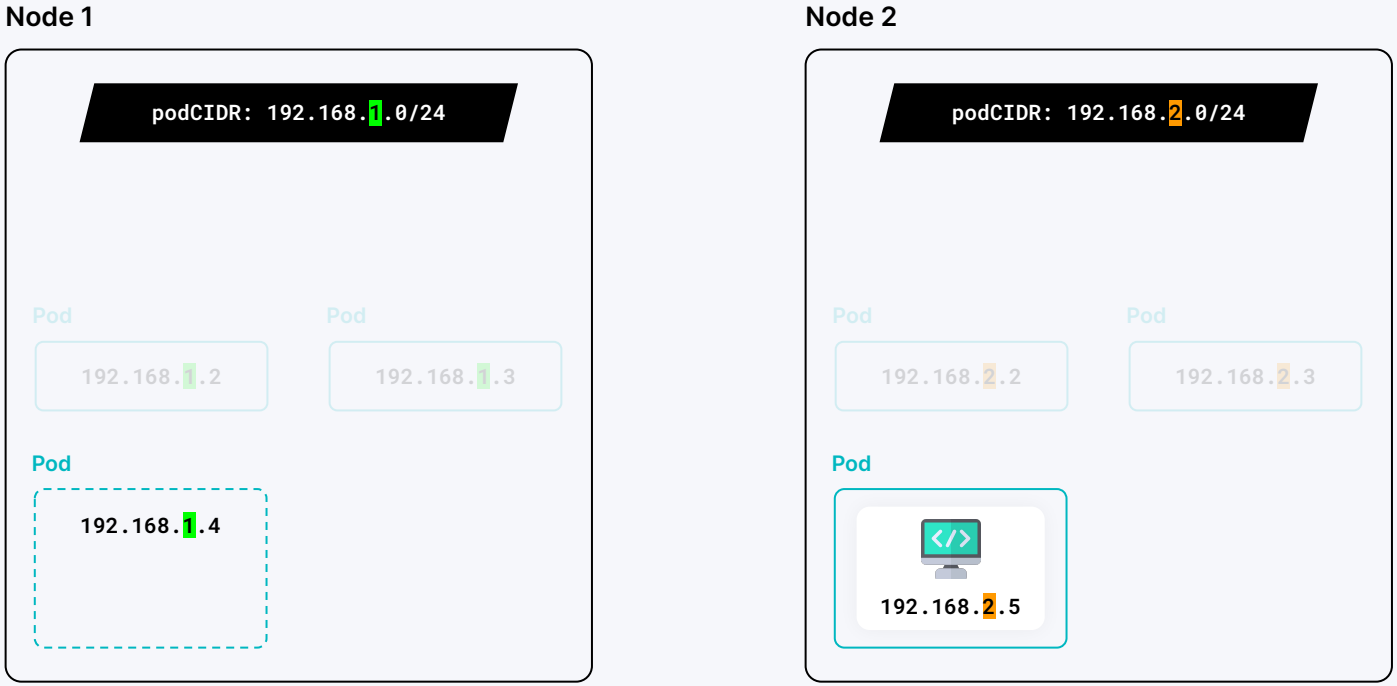
How to make static IPs



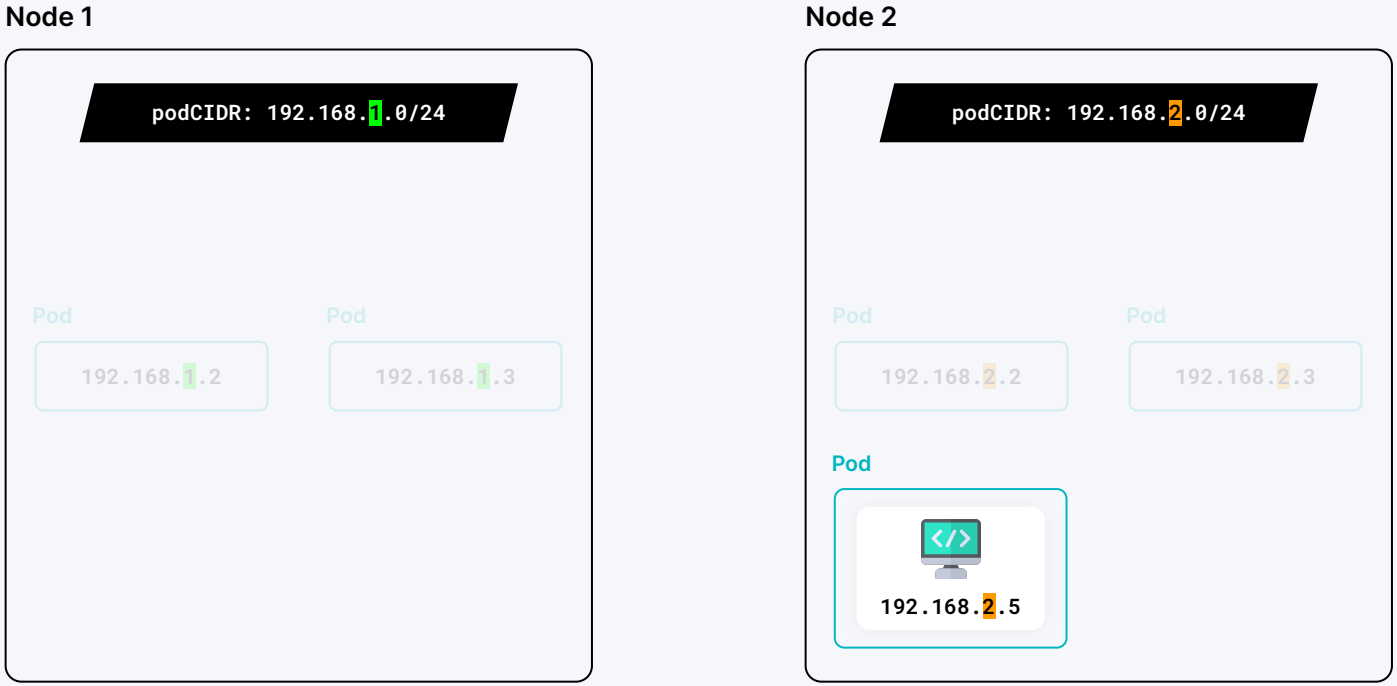
How to make static IPs



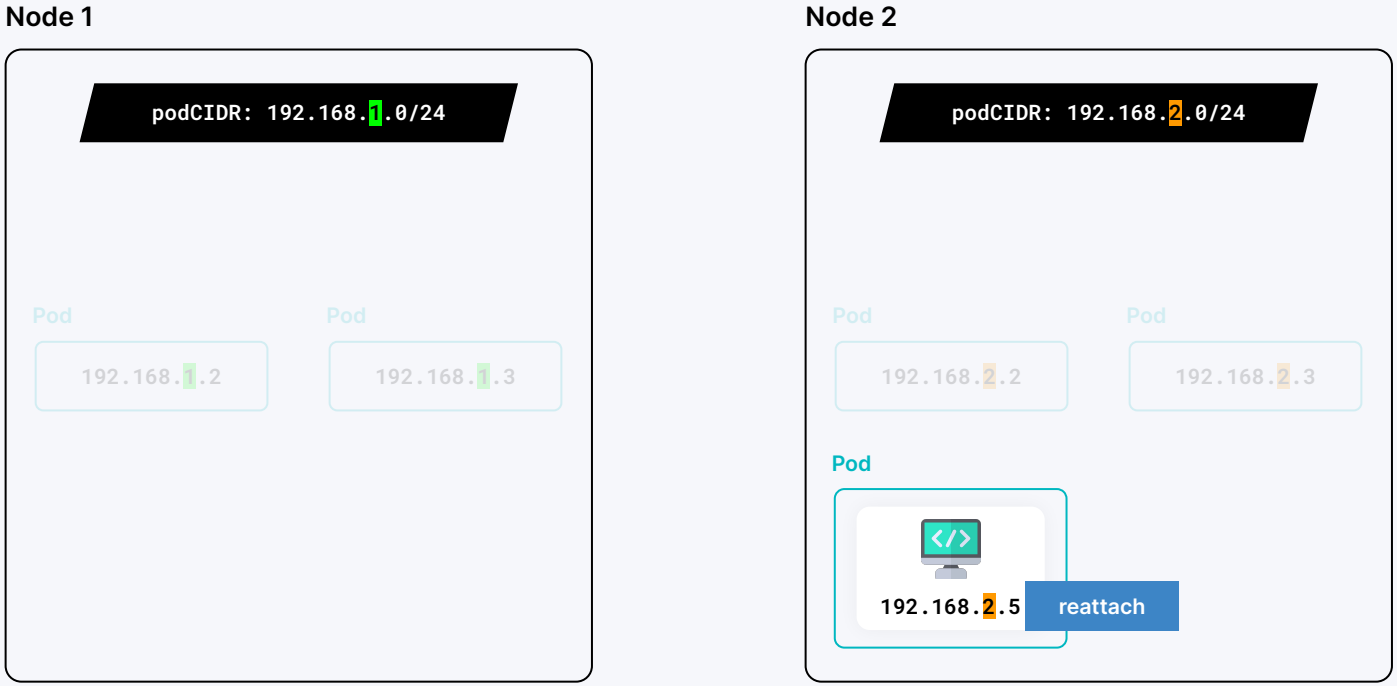
How to make static IPs



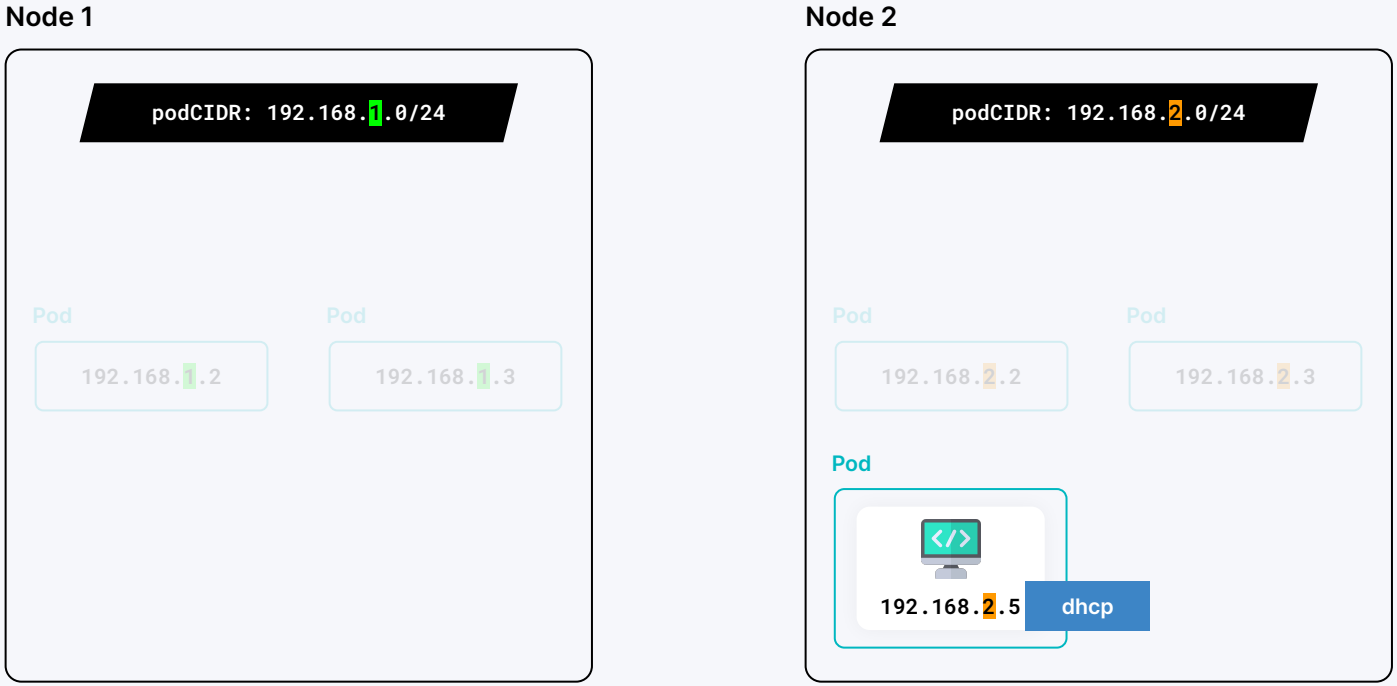
How to make static IPs



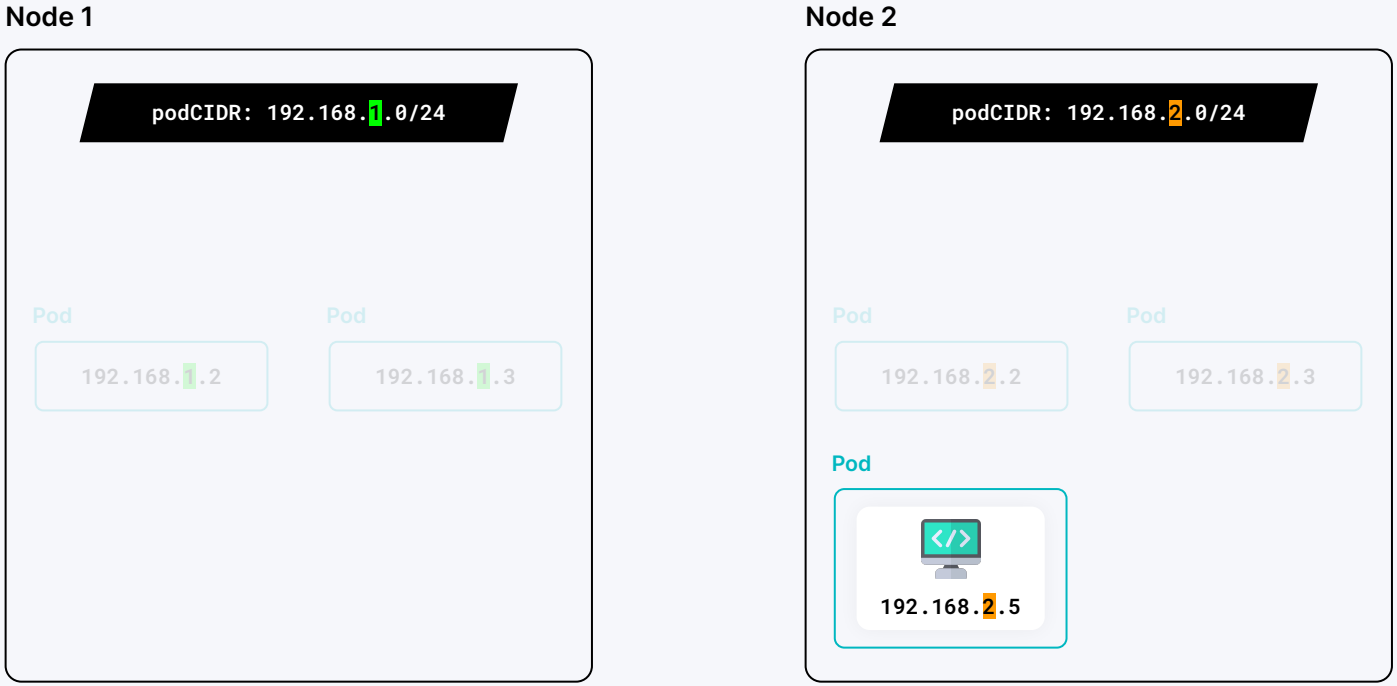
How to make static IPs



How to make static IPs

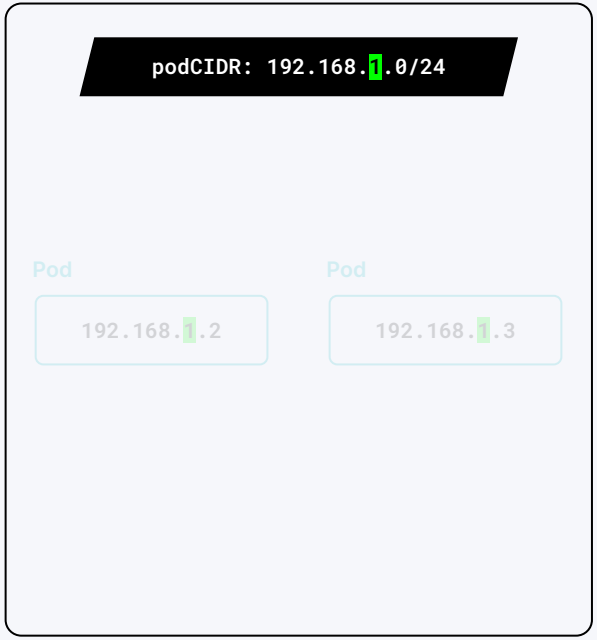


How to make static IPs

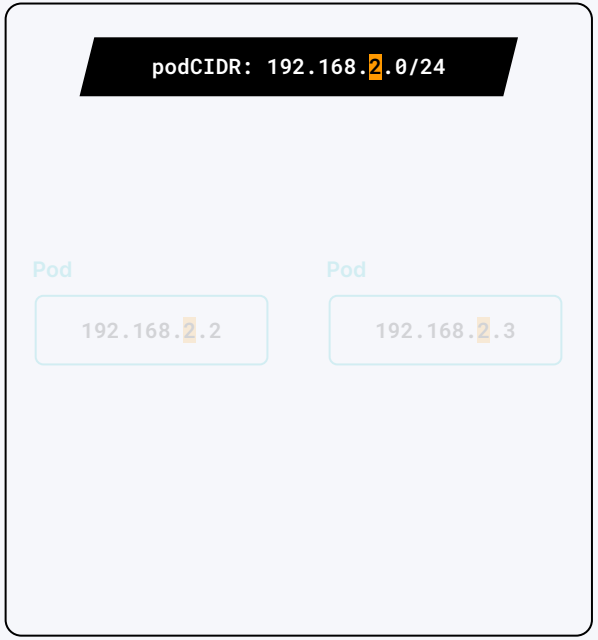


How to make static IPs

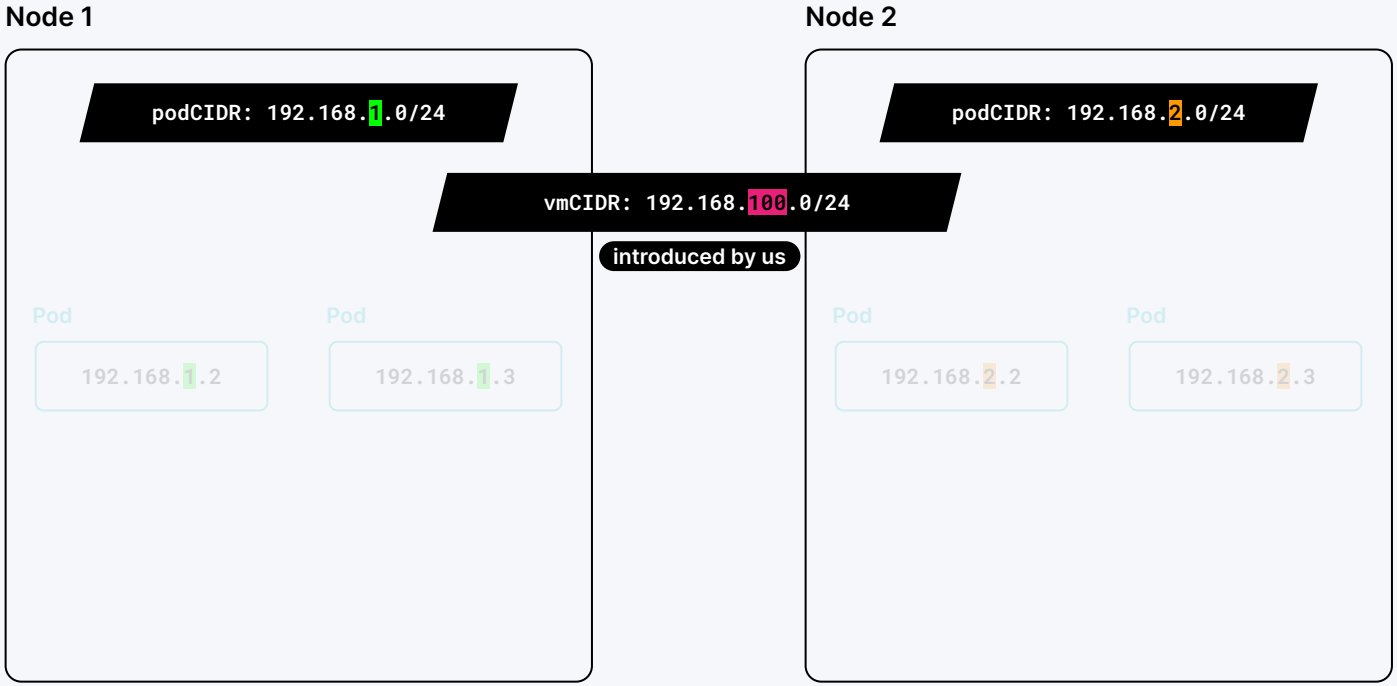
Node 1



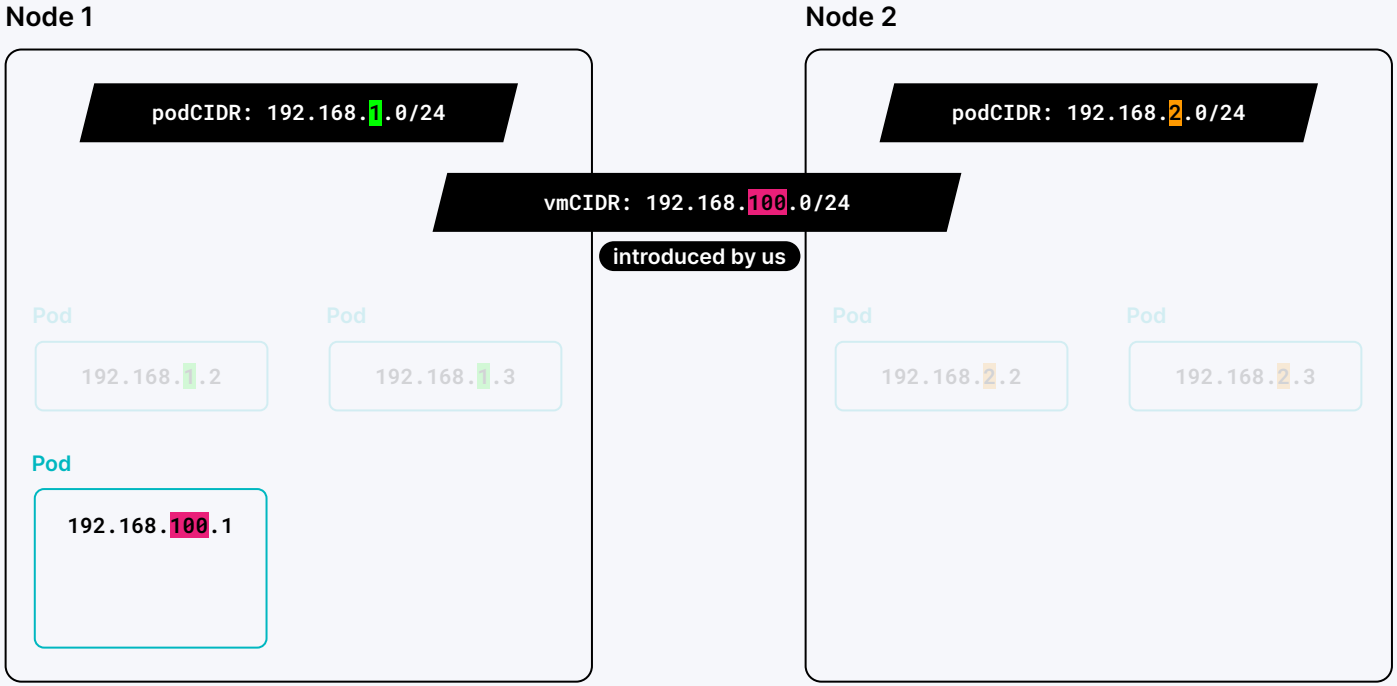
Node 2



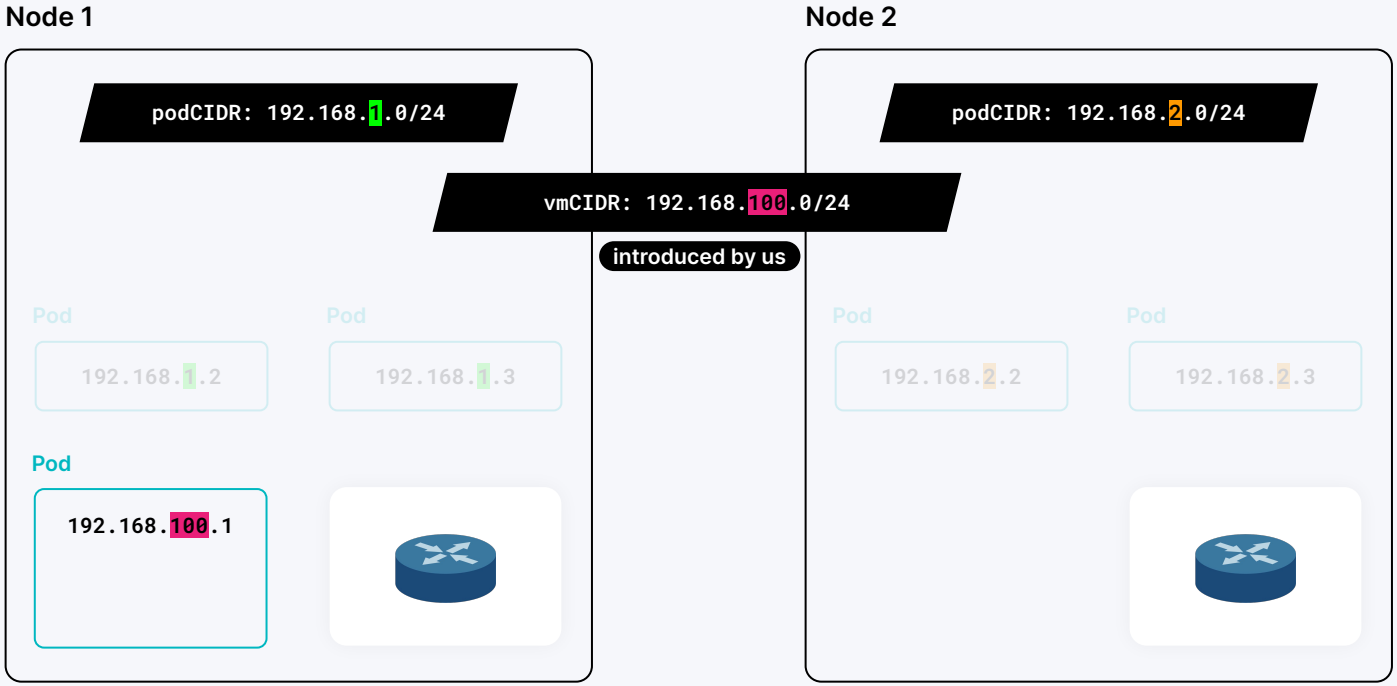
How to make static IPs



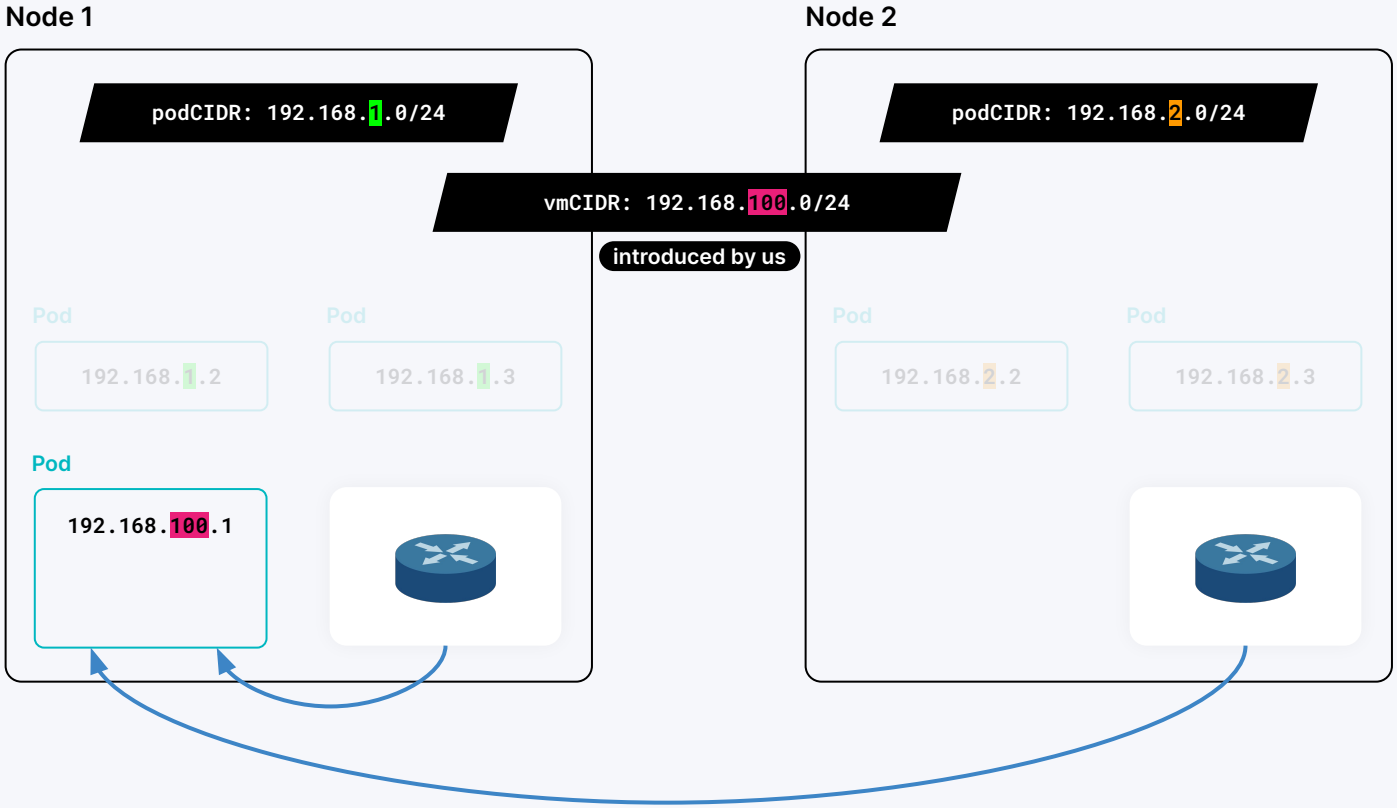
How to make static IPs



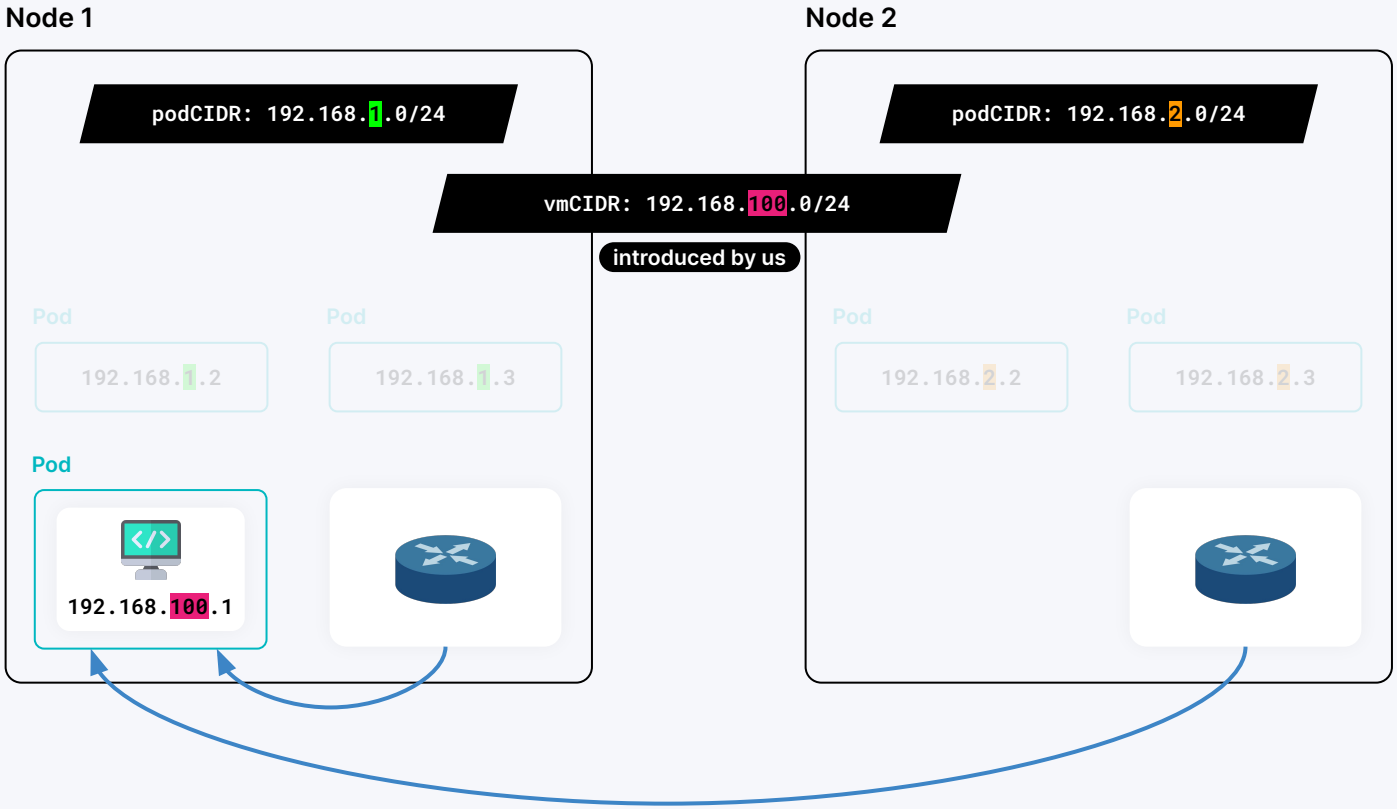
How to make static IPs



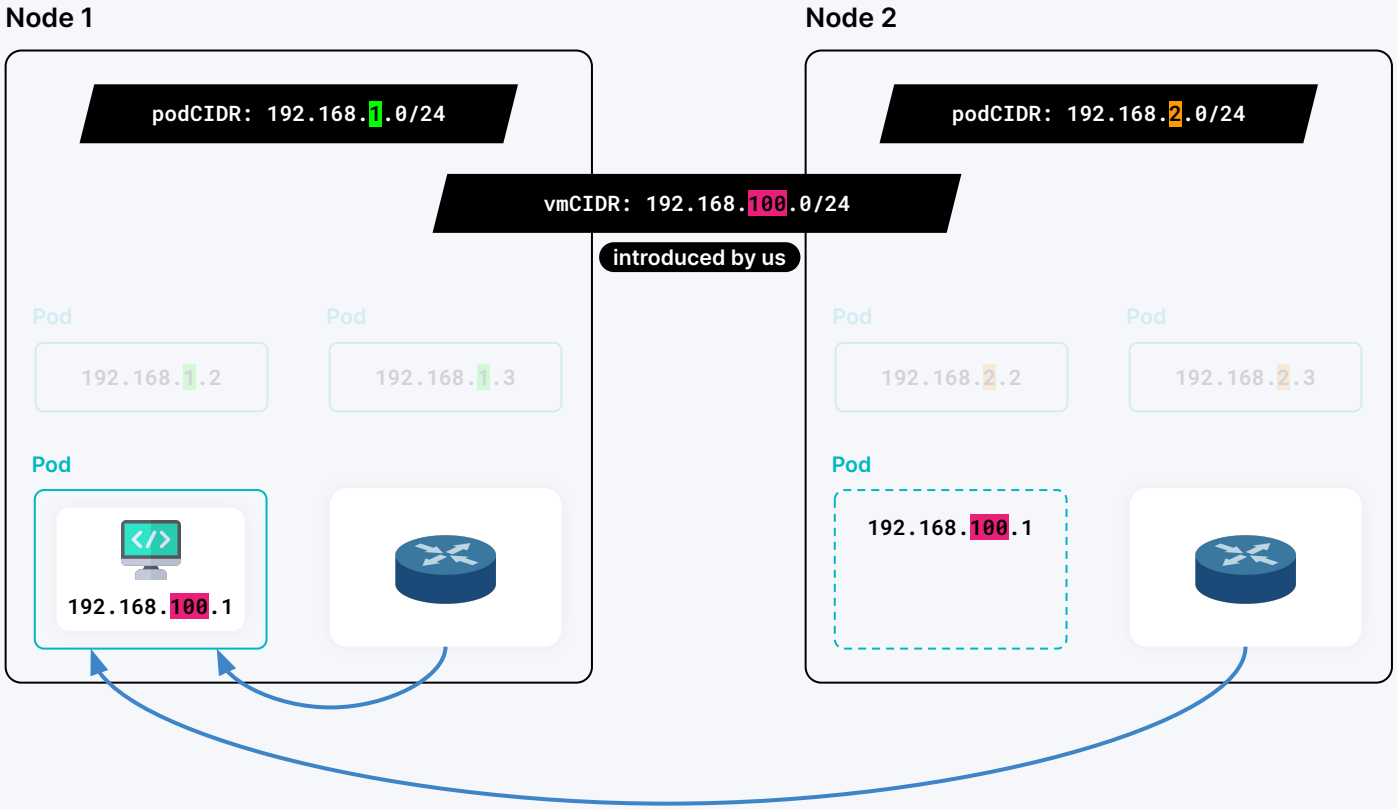
How to make static IPs



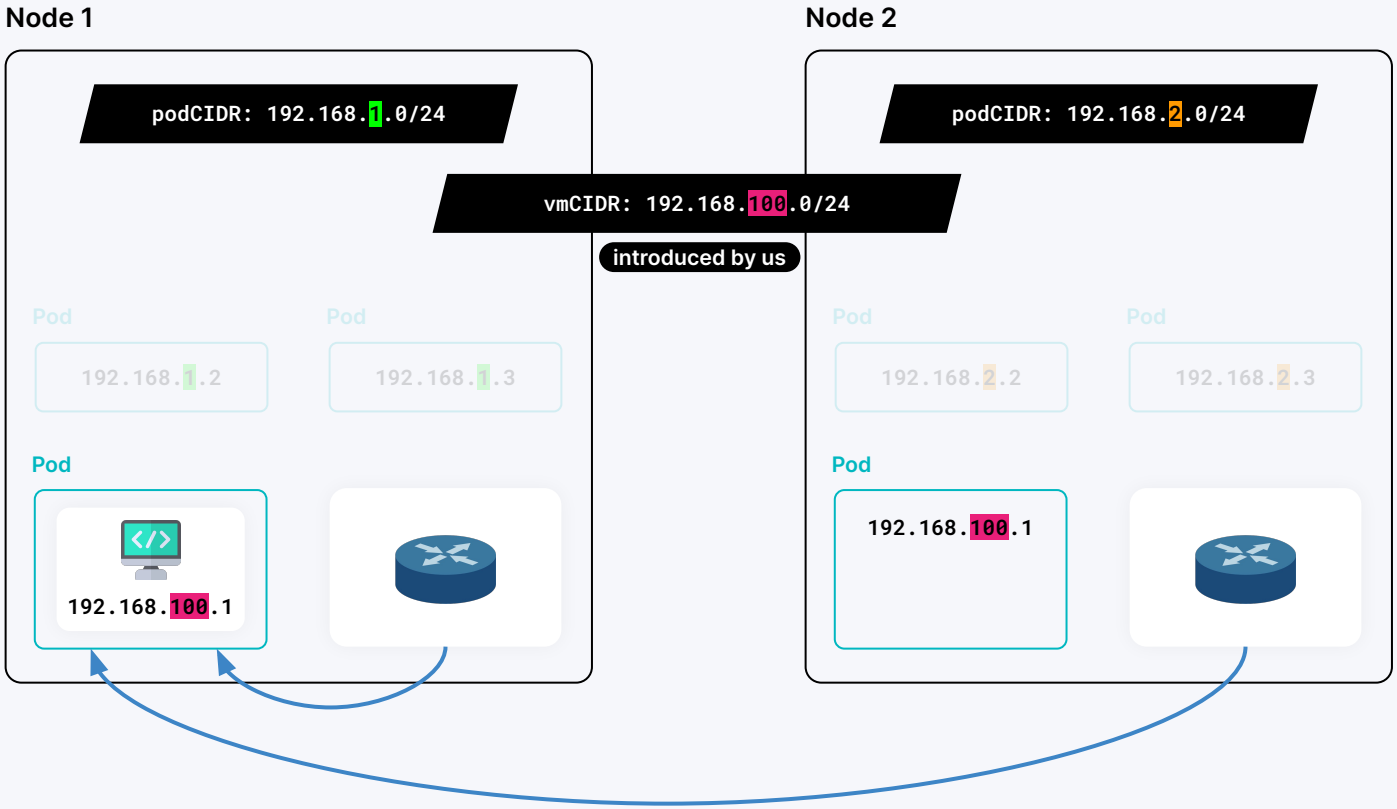
How to make static IPs



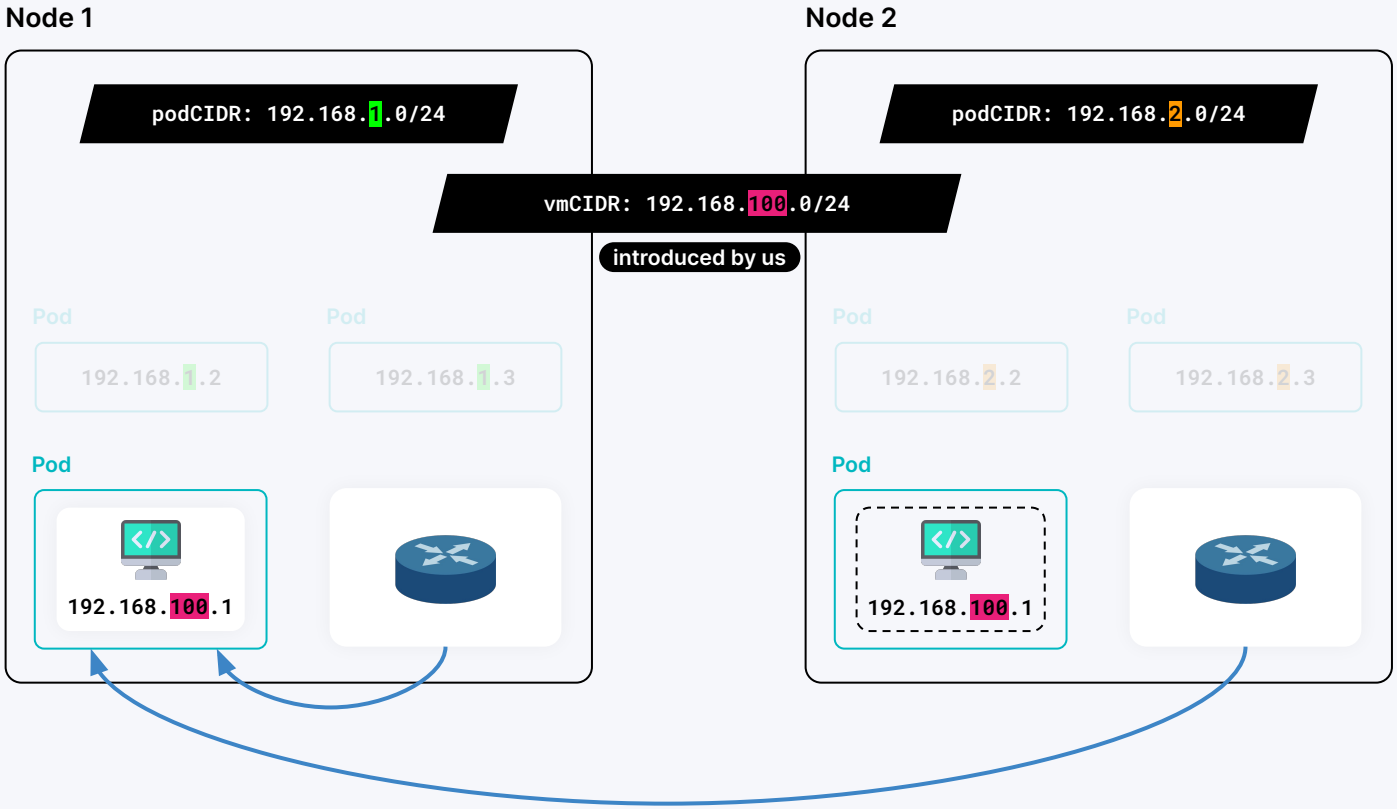
How to make static IPs



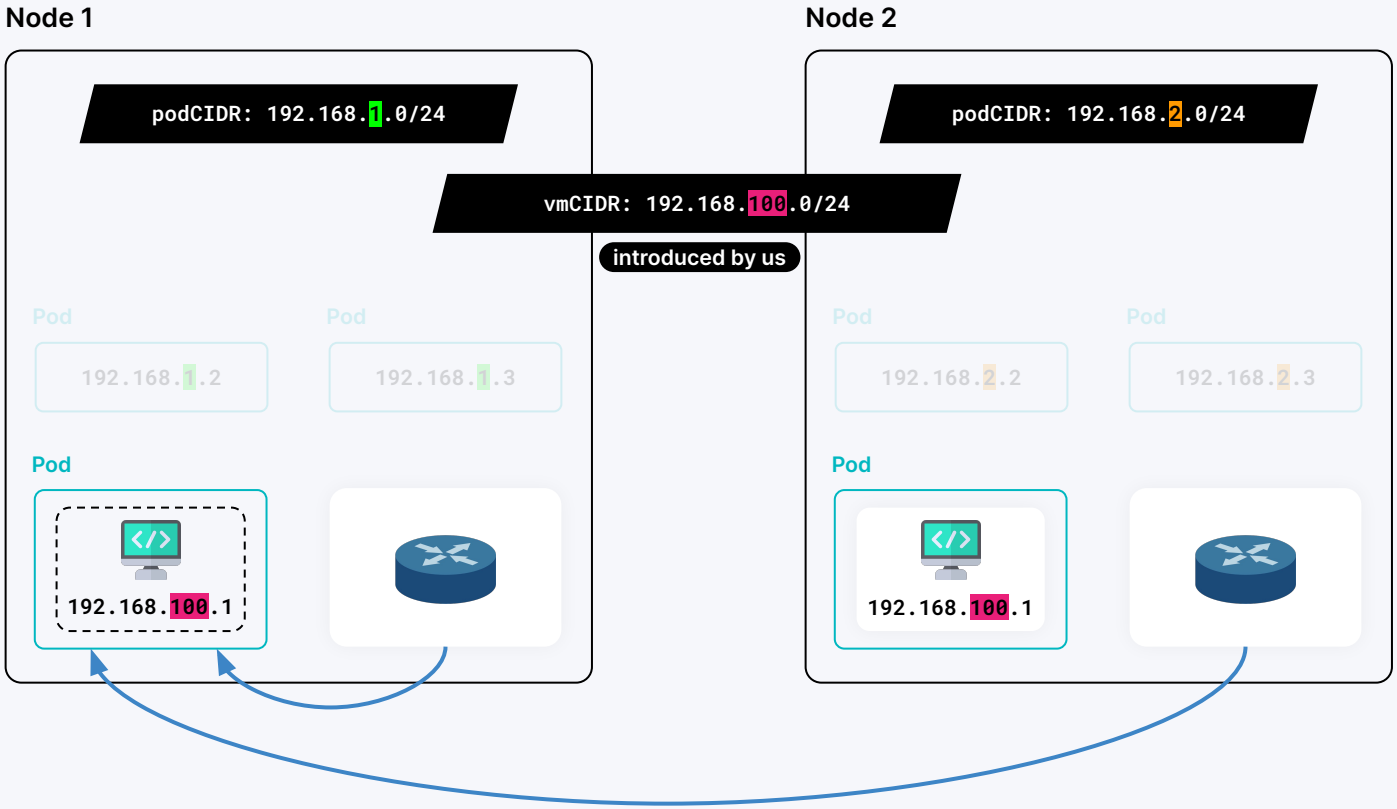
How to make static IPs



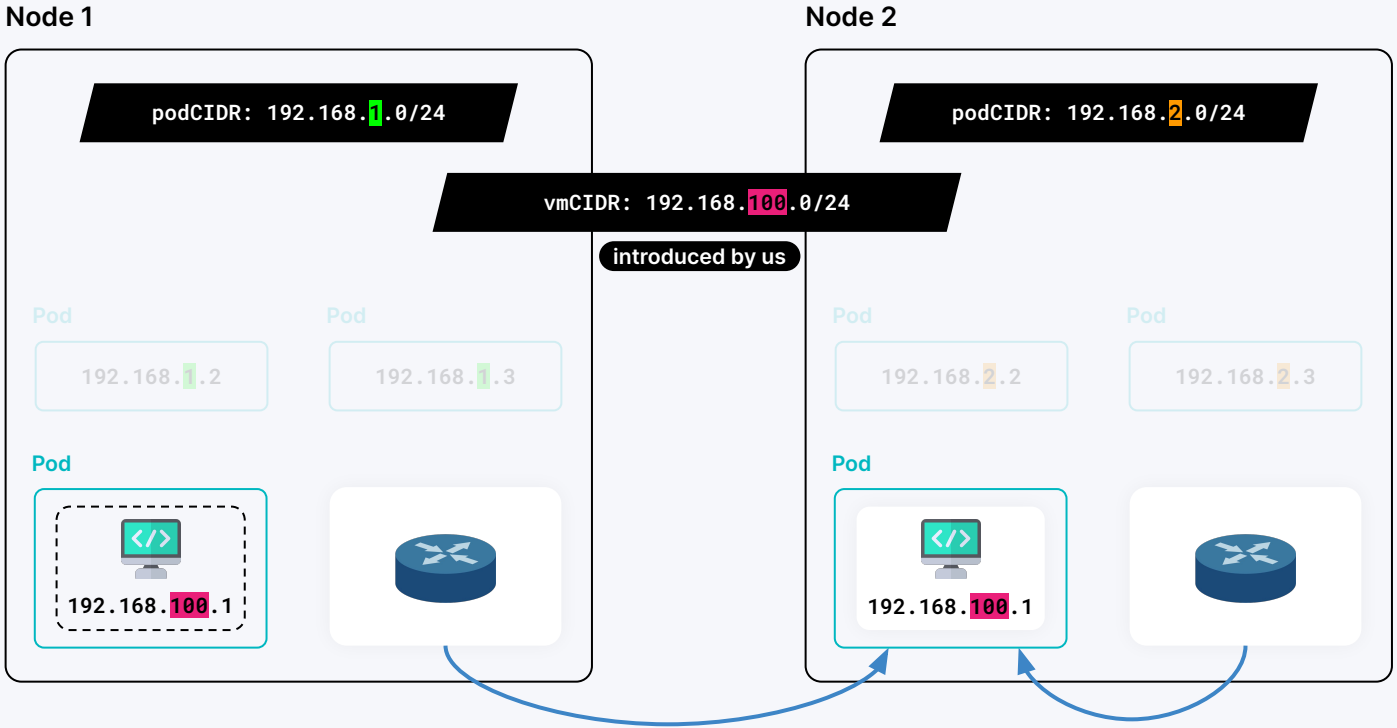
How to make static IPs



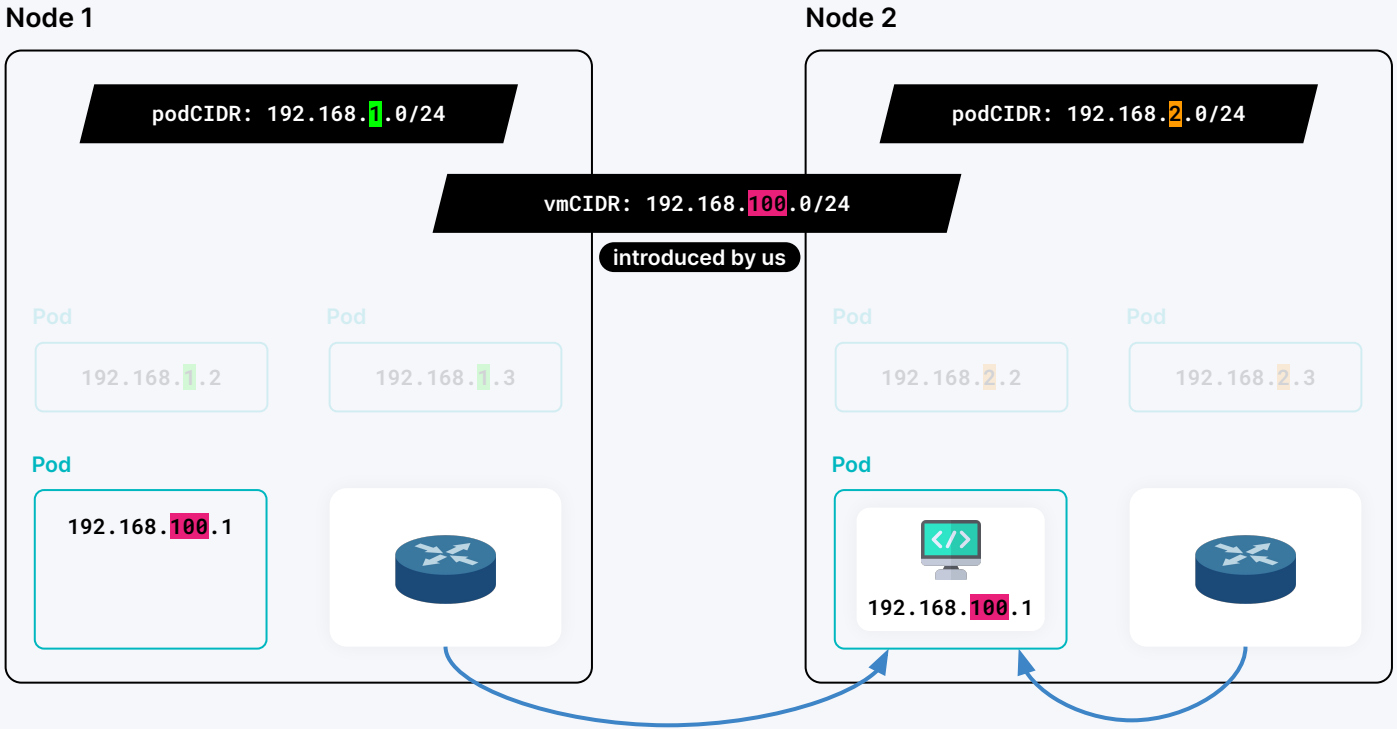
How to make static IPs



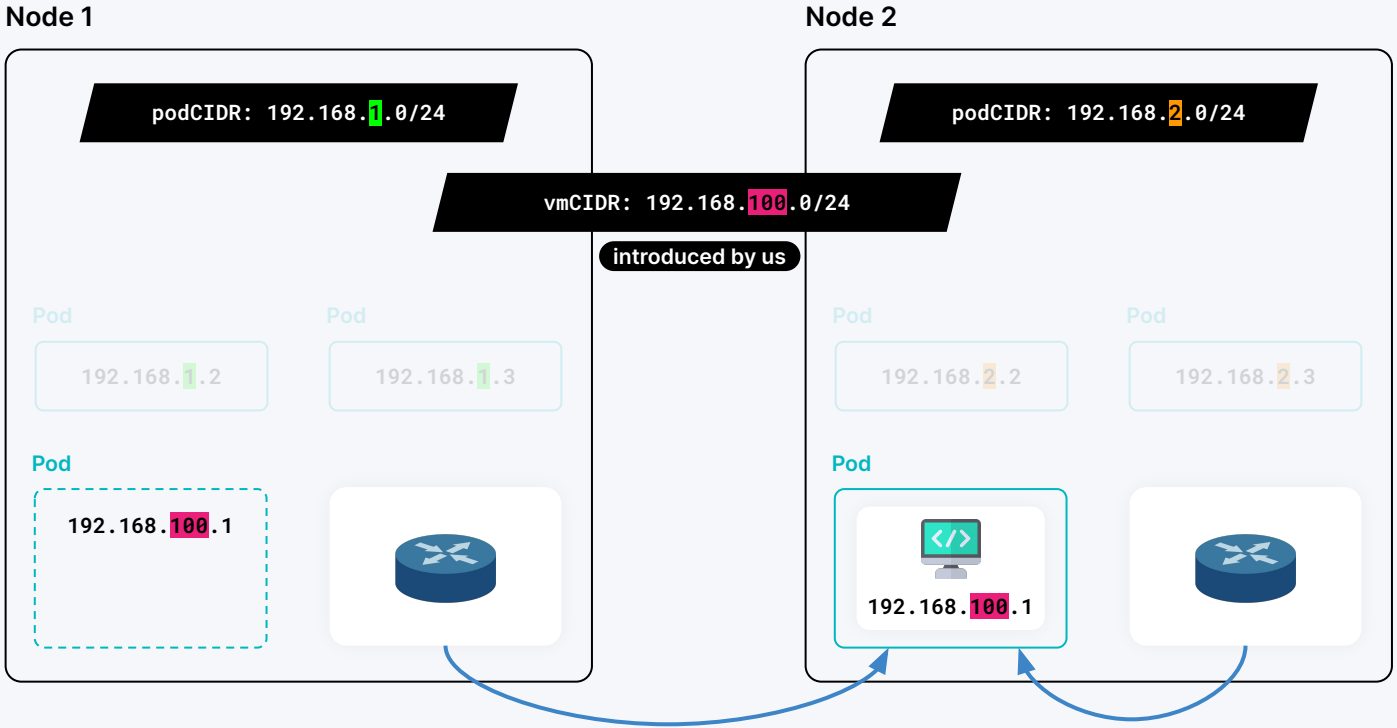
How to make static IPs



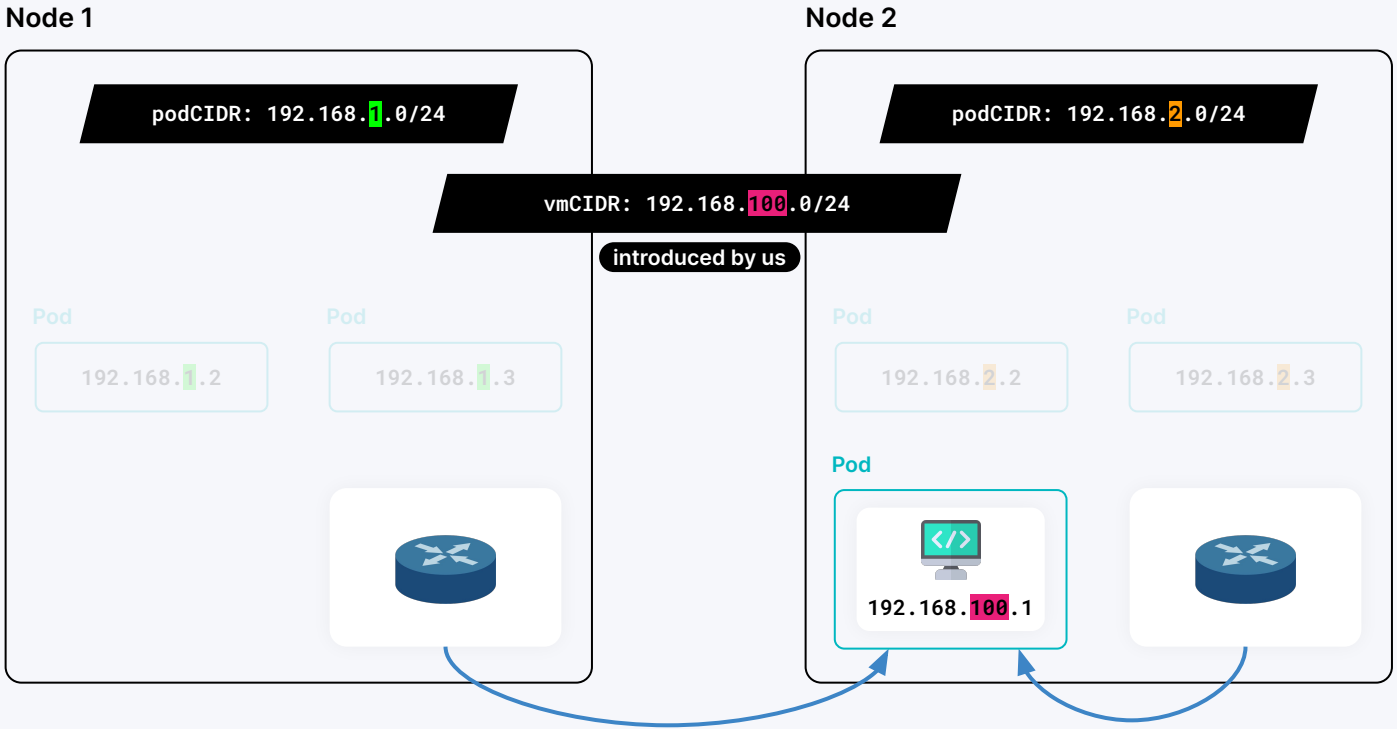
How to make static IPs



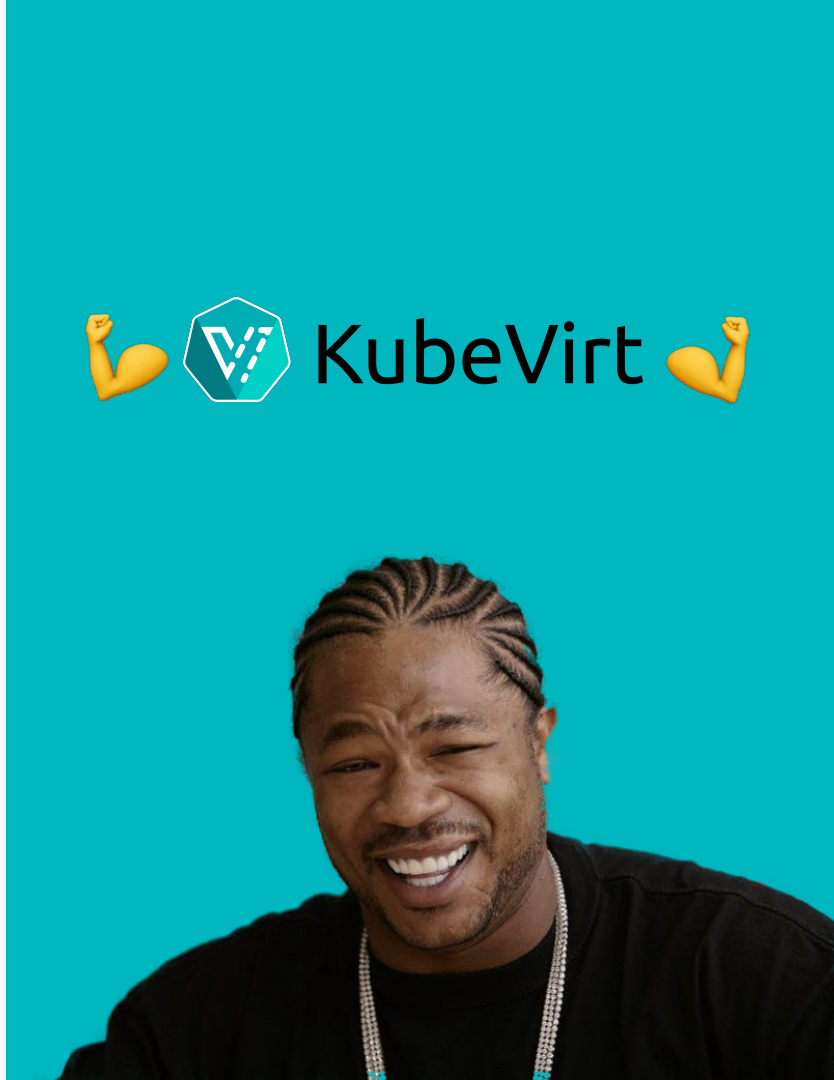
How to make static IPs



How to make static IPs

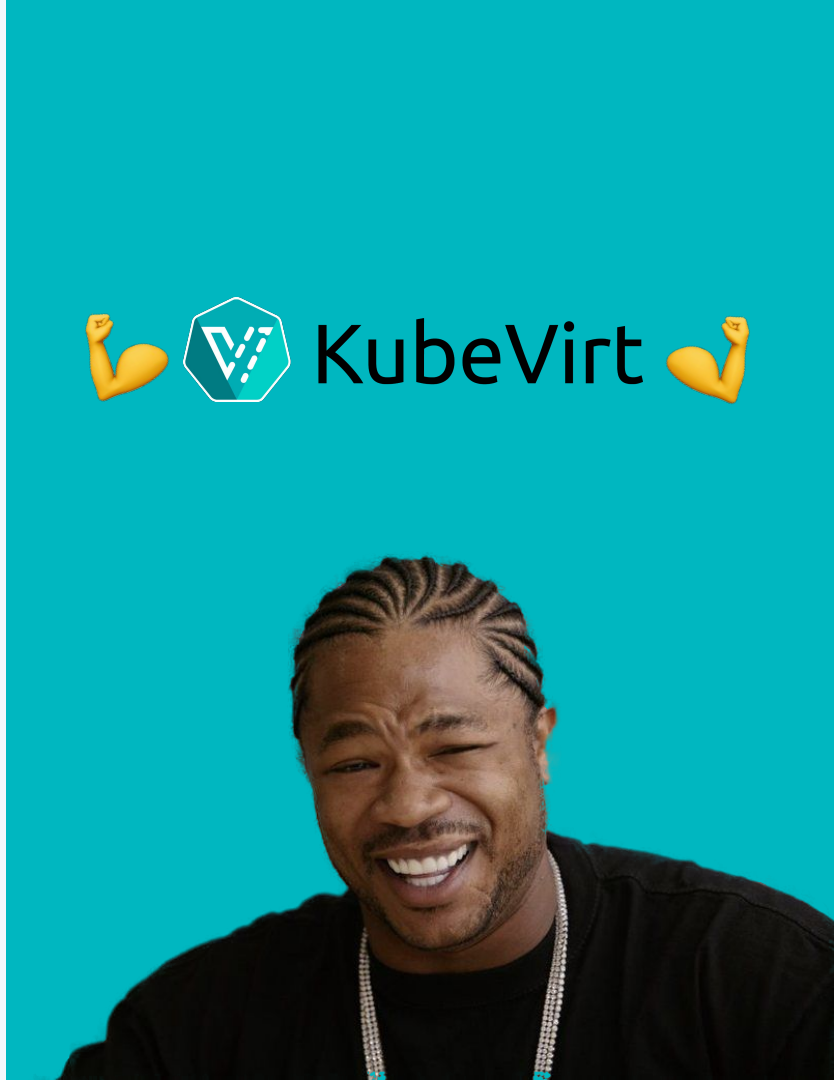


Conclusion



Conclusion

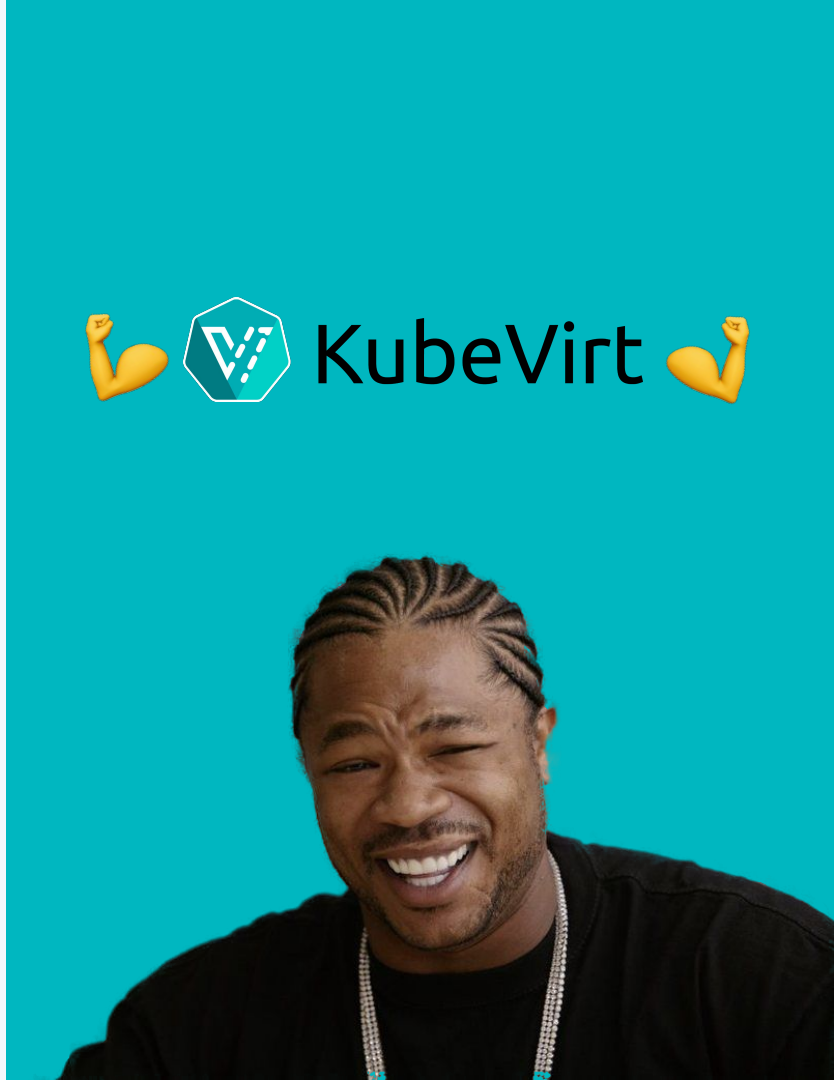
Native **pod networking** instead of Multus



Conclusion

Native **pod networking** instead of Multus

Working cilium **network policies**

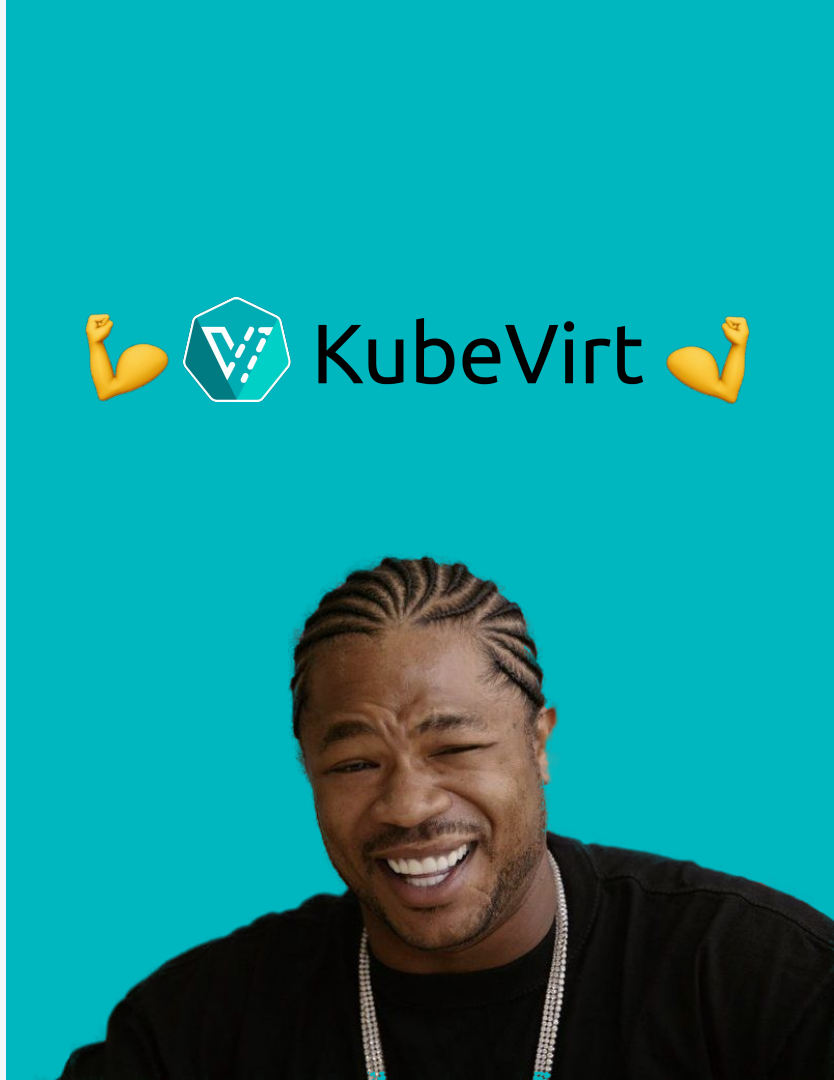


Conclusion

Native **pod networking** instead of Multus

Working cilium **network policies**

Faster **macvtap** binding instead of chain of **bridges**



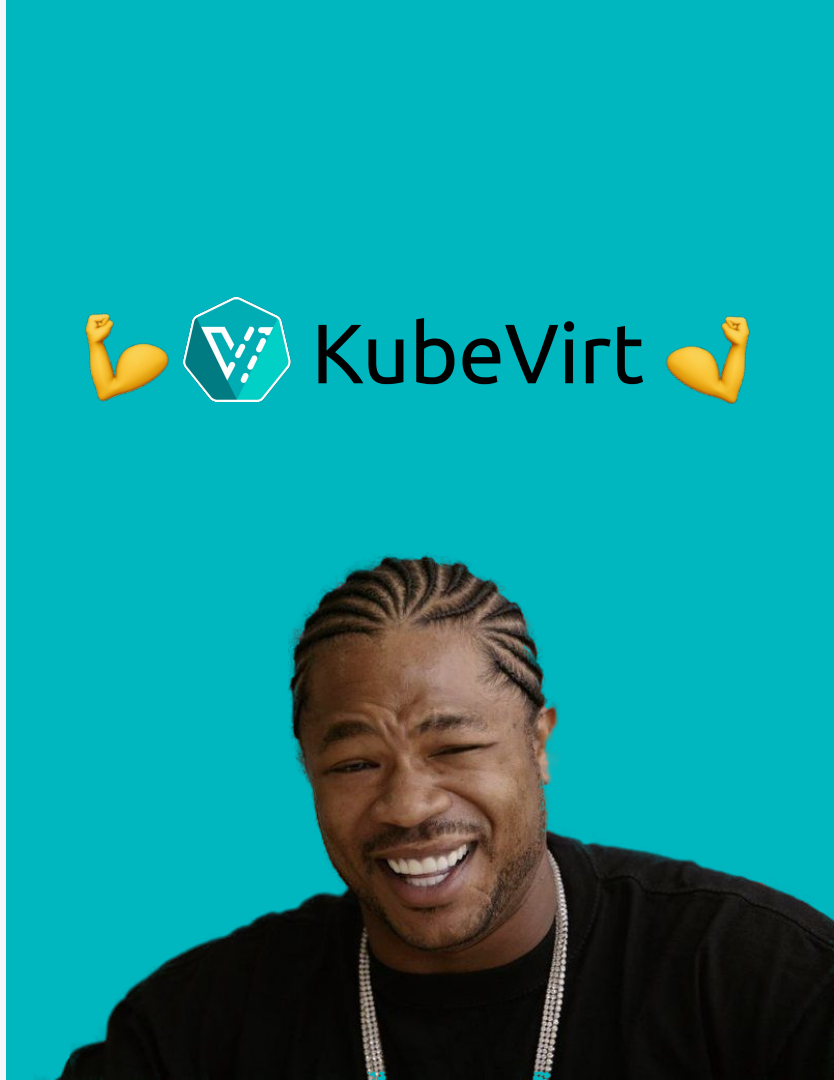
Conclusion

Native **pod networking** instead of Multus

Working cilium **network policies**

Faster **macvtap** binding instead of chain of **bridges**

Working **live-migration** with **static IPs**



Conclusion

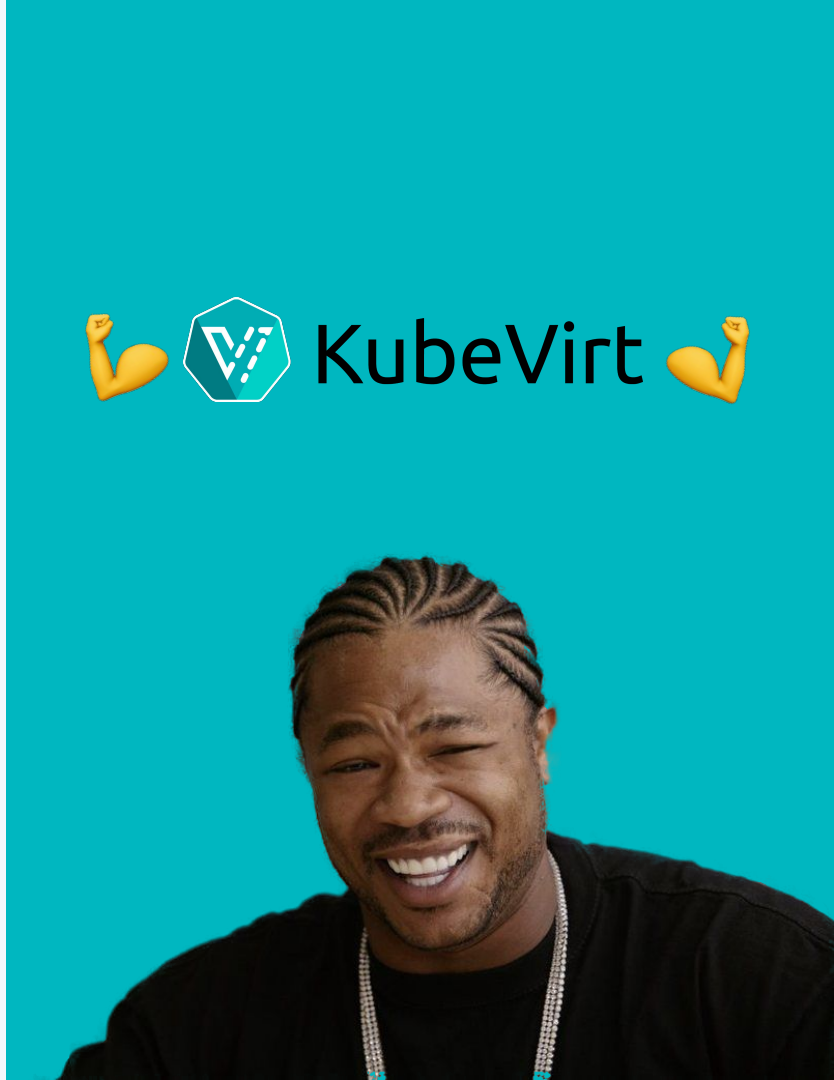
Native **pod networking** instead of Multus

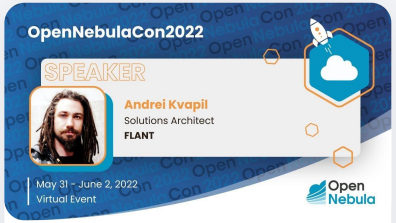
Working cilium **network policies**

Faster **macvtap** binding instead of chain of **bridges**

Working **live-migration** with **static IPs**

Our own **Custom resources** to manage VMs, images and disks





How is cloud computing changing the way of mind

Learn more →

KubeVirt

KubeVirt community channels

Join the discussion →

Proposal: Macvtap mode for pod networking

Support →

Proposal: Live migration for bridged pod network

Support →

Acknowledgements

for assistance with preparing the content

 **Andrey Polovov**

 **Vasily Oleynikov**

 **Yuriy Losev**

for assistance with making the presentation

 **Anton Klimov**

Andrey Kvapil
Solutions Architect

andrey.kvapil@palark.com

PALARK

+49 (0)30 255 555 741
info@palark.com
palark.com

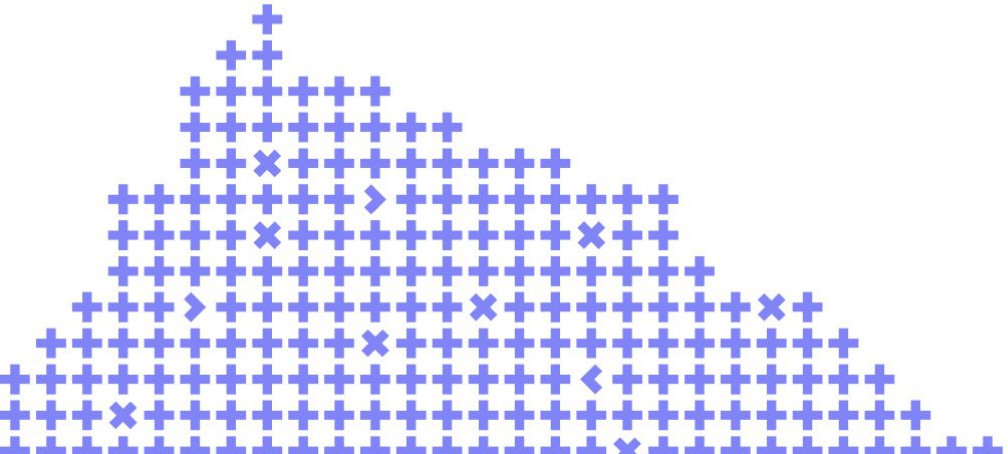
Blog
blog.palark.com

LinkedIn
linkedin.com/company/palark

Twitter
twitter.com/palark_com

Leave your feedback!

You can rate the talk and give a feedback on what you've liked or what could be improved



High Load Armenia

Co-organizer

Yandex